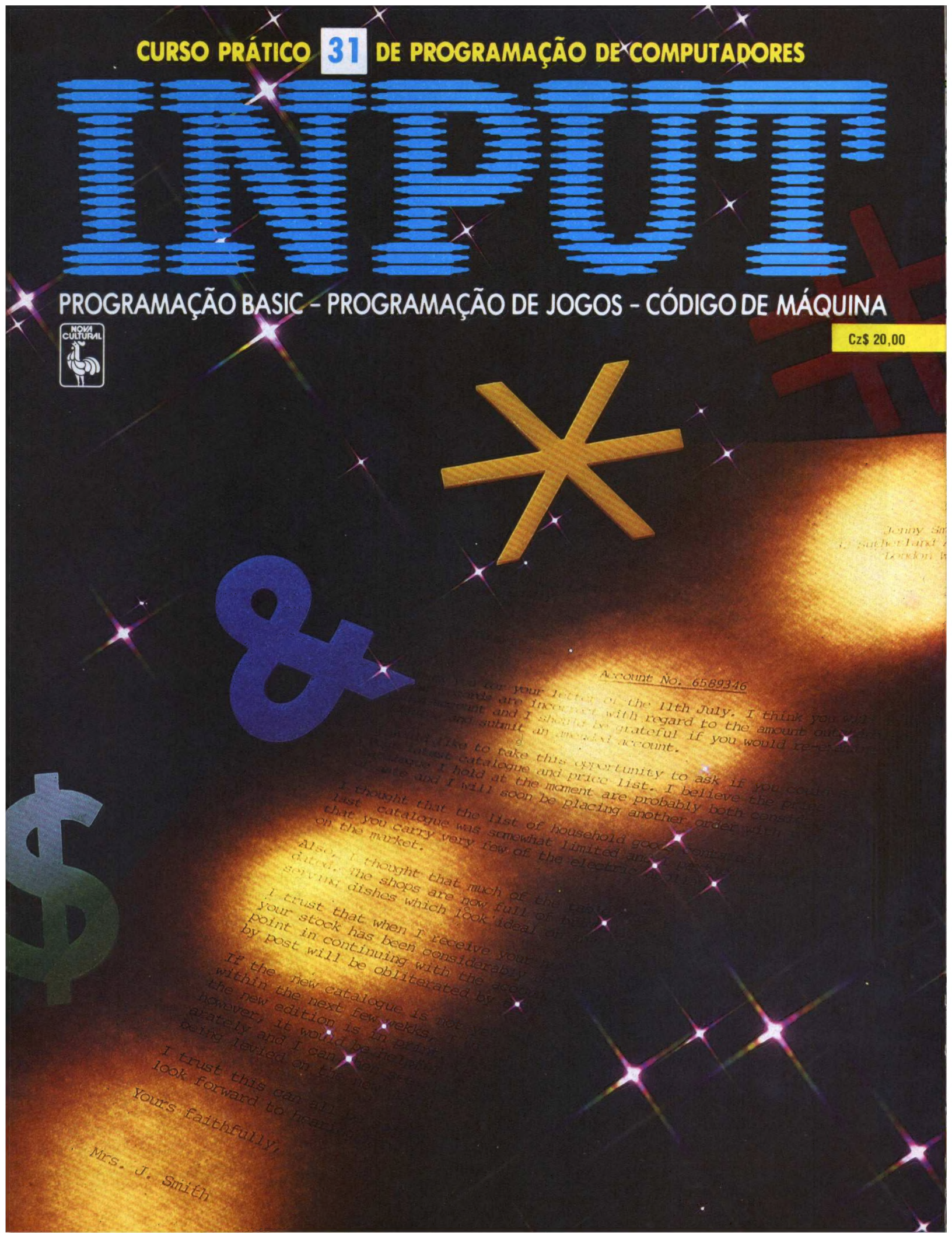
CURSO PRÁTICO 31 DE PROGRAMAÇÃO DE COMPUTADORES

INPUT

PROGRAMAÇÃO BASIC - PROGRAMAÇÃO DE JOGOS - CÓDIGO DE MÁQUINA

NOVA CULTURAL

Cz$ 20,00



INPUT

Vol. 3 Nº 31

NESTE NÚMERO

PROGRAMAÇÃO DE JOGOS

UM SIMULADOR DE VÔO (2)

Ligue as turbinas do avião e levante vôo. Mas não deixe de apertar os cintos, porque... o piloto automático é muito louco. Este artigo continua a explorar as possibilidades de um vôo simulado . 601

PROGRAMAÇÃO BASIC

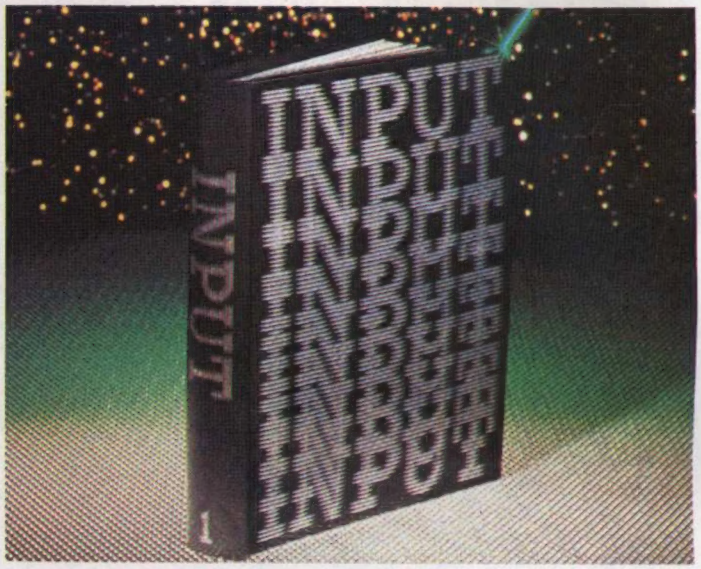
FUNÇÕES SOB ENCOMENDA

As funções sob encomenda facilitam enormemente a resolução de problemas complexos, como calcular juros compostos, transformar letras maiúsculas em minúsculas e encontrar o cubo de um número 608

APLICAÇÕES

UM EDITOR DE TEXTO (3)

A procura incessante da palavra certa tem sido uma verdadeira obsessão para grandes escritores. Este artigo ensina a fazer isso com a prosaica ajuda do computador. E mais: impressão, carta padrão, ajustes à direita e à esquerda, formatação, etc . 614



PLANO DA OBRA

"INPUT" é uma obra editada em fascículos semanais, e cada conjunto de 15 fascículos compõe um volume. A capa para encadernação de cada volume estará à venda oportunamente.

COMPLETE SUA COLEÇÃO

Exemplares atrasados, até seis meses após o encerramento da coleção, poderão ser comprados, a preços atualizados, da seguinte forma: **1. Pessoalmente** — por meio de seu jornaleiro ou dirigindo-se ao distribuidor local, cujo endereço poderá ser facilmente conseguido junto a qualquer jornaleiro de sua cidade. Em São Paulo os endereços são: Rua Brigadeiro Tobias, 773 (Centro); Av. Industrial, 117 (Santo André); e, no Rio de Janeiro: Rua da Passagem, 93 (Botafogo). **2. Por carta** — Poderão ser solicitados exemplares atrasados também por carta, que deve ser enviada para DINAP — Distribuidor Nacional de Publicações — Números Atrasados — Estrada Velha de Osasco, 132 (Jardim Tereza) — CEP 06000 — Osasco — São Paulo. **3. Por telex** — Utilize o nº (011) 33670 ABSA. Em **Portugal**, os pedidos devem ser feitos à Distribuidora Jardim de Publicações Ltd. — Qta. Pau Varais, Azinhaga de Fetais — 2685, Camarate — Lisboa; Tel. 257-2542 — Apartado 57 — Telex 43 069 JARLIS P.
Não envie pagamento antecipado. O atendimento será feito pelo reembolso postal e o pagamento, incluindo as despesas postais, deverá ser efetuado ao se retirar a encomenda na Agência do Correio. **Atenção**: Após seis meses do encerramento da coleção, os pedidos serão atendidos, dependendo da disponibilidade de estoque. **Obs.:** Quando pedir livros, mencione sempre o título e/ou o autor da obra, além do número da edição.

COLABORE CONOSCO

Encaminhe seus comentários, críticas, sugestões ou reclamações ao Serviço de Atendimento ao Leitor — Caixa Postal 9442, São Paulo — SP.

Editor
VICTOR CIVITA

REDAÇÃO
Diretora Editorial: Iara Rodrigues

Editor Executivo: Antonio José Filho

Editor Chefe: Paulo de Almeida
Editor de Texto: Cláudio A. V. Cavalcanti
Chefe de Arte: Carlos Luiz Batista
Assistentes de Arte: Ailton Oliveira Lopes, Dilvacy M. Santos, Grace Alonso Arruda, José Maria de Oliveira, Monica Lenardon Corradi
Secretária de Redação/Coordenadora: Stefania Crema
Secretários de Redação: Beatriz Hagström, José Benedito de Oliveira Damião, Maria de Lourdes Carvalho, Marisa Soares de Andrade, Mauro de Queiroz

COLABORADORES
Consultor Editorial Responsável: Dr. Renato M. E. Sabbatini (Diretor do Núcleo de Informática Biomédica da Universidade Estadual de Campinas)
Execução Editorial: DATAQUEST Assessoria em Informática Ltda., Campinas, SP
Tradução: Reinaldo Cúrcio
Tradução, adaptação, programação e redação: Abílio Pedro Neto, Aluísio J. Dornellas de Barros, Marcelo R. Pires Therezo, Raul Neder Porrelli
Coordenação Geral: Rejane Felizatti Sabbatini
Editora de Texto: Ana Lúcia B. de Lucena
Assistente de Arte: Dagmar Bastos Sampaio

COMERCIAL
Diretor Comercial: Roberto Martins Silveira
Gerente Comercial: Flávio Maculan
Gerente de Circulação: Denise Maria Mozol

PRODUÇÃO
Gerente de Produção: João Stungis
Coordenador de Impressão: Atilio Roberto Bonon

Preparador de Texto/Coordenador: Eliel Silveira Cunha
Preparadores de Texto: Alzira Moreira Braz, Ana Maria Dilguerian, Karina Ap. V. Grechi, Levon Yacubian, Luciano Tasca, Maria Teresa Galluzzi, Maria Teresa Martins Lopes, Paulo Felipe Mendrone
Revisor/Coordenador: José Maria de Assis
Revisoras: Conceição Aparecida Gabriel, Isabel Leite de Camargo, Ligia Aparecida Ricetto, Maria de Fátima Cardoso, Nair Lúcia de Brito
Paste-up: Anastase Potaris, Balduino F. Leite, Edson Donato

© Marshall Cavendish Limited, 1984/85.
© Editora Nova Cultural Ltda., São Paulo, Brasil, 1986.

Edição organizada pela Editora Nova Cultural Ltda.
Av. Brigadeiro Faria Lima, nº 2000 - 3º andar
CEP 01452 - São Paulo - SP - Brasil
(Artigo 15 da Lei 5 988, de 14/12/1973).
Esta obra foi composta na AM Produções Gráficas Ltda. e impressa na Divisão Gráfica da Editora Abril S.A.

UM SIMULADOR DE VÔO (2)

- UM PILOTO AUTOMÁTICO PARA CONDUZIR O AVIÃO
- APROXIME-SE DA PISTA
- COMO TRAÇAR O RUMO
- FAÇA O PAINEL FUNCIONAR

Ligue as turbinas do avião e observe o jogo dos ponteiros no painel dos mostradores. Mas não deixe de apertar o cinto, pois o piloto automático é meio maluco.

Já vimos como reproduzir o interior de uma cabine de avião na tela do microcomputador. No programa do artigo anterior para as linhas TRS-Color, MSX, Apple e TK-2000, o avião foi deixado estático, em pleno ar. Evidentemente, uma situação assim nunca ocorre na vida real.

Neste artigo, nosso aeroplano vai sair voando e os instrumentos do painel vão ganhar vida, para que possamos acompanhar seu movimento, apesar de ainda não estarmos em condições de mantê-lo sob controle.

COMO FAZER O AEROPLANO VOAR

Esta é certamente a parte mais longa da listagem: uma série complexa de variáveis independentes deve ser constantemente calculada para reproduzir e controlar o comportamento do aparelho em vôo. Ao mesmo tempo, os instrumentos do painel precisam ser redesenhados à medida que a posição e a altitude do avião sofrem modificações. Da mesma forma, os números dos mostradores são impressos para que saibamos sua posição e sua orientação.

APROXIME-SE DA PISTA

Uma imagem da pista no radar nos mostra o ângulo de aproximação no Spectrum e no MSX. No TRS-Color, uma imagem da pista pode ser observada através da janela, à medida que chegamos próximo à pista. Esta aparece progressivamente, aproveitando a capacidade do TRS-Color em desenhar elipses. Já os usuários do Apple e do TK-2000 devem se orientar pela bússola (*bearing*), pela direção da pista (*runway*) e pela distância do avião em relação ao eixo longitudinal da pista (*drift*).

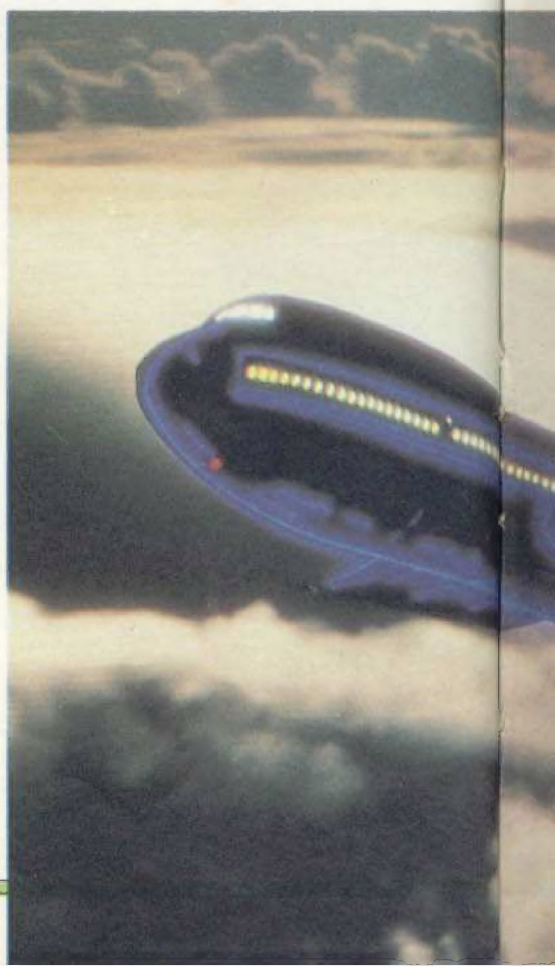
O CURSO DO APARELHO

Diversos fatores devem ser considerados para que se possa calcular a posição do aparelho num dado momento. A direção em que estamos seguindo, por exemplo, é afetada pela dos ventos e pela posição dos ailerons instalados nas duas asas. A velocidade de cruzeiro depende, por sua vez, da intensidade do vento. As velocidades de ascensão e de mergulho estão sempre relacionadas

com a de cruzeiro, e assim por diante.

Para manter sempre atualizados os valores nos mostradores, as variáveis têm de ser recalculadas e colocadas novamente na tela.

```
   2 LET WY=0: LET WX=0: LET GZ
   =0: LET GY=0: LET GX=0
   5 LET RW=0: LET Y1=120: LET
   Y2=120: LET Y3=40: LET Y4=40
   : LET POW=0: LET GC=0: LET R
   B=0: LET LL=0: LET YC=0: LET
   AD=0: LET ST=0: LET RL=0:
   LET BC=0: LET NC=0: LET PT=0
   : LET PX=0: LET VZ=0: LET VY
   =0: LET VX=0
 500 LET RA=AD*C: LET VX=AS*SIN
 RA
 510 LET VY=AS*COS RA: RETURN
1000 LET PZ=PZ+GZ: LET PY=PY+GY
: LET PX=PX+GX
1025 IF ST=1 THEN  PRINT  OVER
1;AT 4,9;"M E R G U L H O": LET
 ST=0: GOTO 1040
1030 IF AS<30 THEN  GOSUB 1500
1040 LET AD=AD+RL: IF AD<0 THEN
  LET AD=AD+360
1050 IF AD>359 THEN  LET AD=AD-
360
1060 LET VZ=AS*SIN (PT*C)-10+AS
/15
1070 LET GZ=VZ: LET GY=VY+WY: L
ET GX=VX+WX
1080 IF VY=0 THEN  LET GD=-PI/2
: GOTO 1100
1090 LET GD=-ATN (VX/VY)/C
1100 GOSUB 500
1110 RETURN
1500 LET ST=1: PRINT  OVER 1;AT
 4,9;"M E R G U L H O": FOR M=1
 TO 4: FOR N=20 TO -20 STEP -4:
 SOUND .01,N: NEXT N: NEXT M
1510 LET RL=INT (RND*21)-9: LET
 PT=-21-INT (RND*5)
1520 RETURN
2180 IF GC<>0 THEN  GOSUB 2200
2190 LET AS=AS+16*(TC*30-AS-8*P
T)/AS: GOSUB 2200: GOTO 2205
2200 PLOT 35,50: DRAW  OVER 1;1
5*SIN (AS*PI/200),15*COS (AS*PI
/200): RETURN
2205 IF GC<>0 THEN  PLOT 155,50
: DRAW  OVER 1;10*SIN (TN*PI/5)
,10*COS (TN*PI/5): PLOT 155,50:
 DRAW  OVER 1;15*SIN (UN*PI/500
),15*COS (UN*PI/500)
2210 LET TN=PZ/1000: LET UN=PZ-
1000*INT TN: PLOT 155,50: DRAW
 OVER 1;10*SIN (TN*PI/5),10*COS
 (TN*PI/5): PLOT 155,50: DRAW
OVER 1;15*SIN (UN*PI/500),15*CO
S (UN*PI/500)
2220 IF GC<>0 THEN  GOSUB 2230
2225 IF POW=-1 AND TC>2 THEN  L
ET TC=TC-2
2226 IF POW=1 AND TC<8.8 THEN
LET TC=TC+2
2228 GOSUB 2230: GOTO 2240
2230 PLOT 215,50: DRAW  OVER 1;
15*SIN (TC*PI/5),15*COS (TC*PI/
5): RETURN
2240 PRINT AT 21,2;ABS INT AD;"
  "
2250 IF PY=0 THEN  LET RB=0: GO
TO 2260
2255 LET RB=ATN (PX/PY)/C: IF P
Y>0 THEN  LET RB=RB+180
2260 IF RB<0 THEN  LET RB=RB+36
0
2270 PRINT AT 21,10;INT RB;"  "
;AT 21,18;ABS INT PX;"  "
2280 PRINT AT 21,25;INT (SQR (P
Y*PY+PX*PX));" "
2290 IF (Y1<=110 AND Y2<=110) O
R (Y1>=130 AND Y2>=130) THEN  G
OTO 2300
2295 IF GC<>0 THEN  PLOT  OVER
1;X1,168-Y1: DRAW  OVER 1;X2-PE
EK 23677,168-Y2-PEEK 23678
2300 LET YC=120+(PT/3): LET X1=
80: LET X2=110: LET Y1=YC+17*TA
N (RL*2*C): LET Y2=YC-17*TAN (R
L*2*C)
2310 IF (YC<110 OR YC>130) AND
RL=0 THEN  GOTO 2376
2320 IF Y1<110 THEN  LET X1=95-
(95-X1)*(110-YC)/(Y1-YC): LET Y
1=110: GOTO 2340
2330 IF Y1>130 THEN  LET X1=95-
(95-X1)*(130-YC)/(Y1-YC): LET Y
1=130
2340 IF Y2<110 THEN  LET X2=95-
(95-X2)*(110-YC)/(Y2-YC): LET Y
2=110: GOTO 2360
2350 IF Y2>130 THEN  LET X2=95-
(95-X2)*(130-YC)/(Y2-YC): LET Y
2=130
2360 IF X1<80 OR X2>110 THEN  G
OTO 2376
2370 PLOT  OVER 1;X1,168-Y1: DR
AW  OVER 1;X2-PEEK 23677,168-Y2
-PEEK 23678
2376 IF (RL=RR AND PP=PT) THEN
 GOTO 2500
2377 IF (Y3<=2 AND Y4<=2) OR (Y
3>=90 AND Y4>=90) THEN  GOTO 23
80
2378 IF GC<>0 THEN  PLOT  OVER
1;X3,176-Y3: DRAW  OVER 1;X4-PE
EK 23677,(176-Y4)-PEEK 23678
2380 LET YC=33+PT*4: LET X3=11:
 LET X4=244: LET Y3=YC+118*TAN
(RL*2*C): LET Y4=YC-118*TAN (RL
*2*C)
2390 IF (YC<2 OR YC>90) AND RL=
0 THEN  GOTO 2450
2400 IF Y3<2 THEN  LET X3=128-(
128-X3)*(2-YC)/(Y3-YC): LET Y3=
2: GOTO 2420
2410 IF Y3>90 THEN  LET X3=128-
(128-X3)*(90-YC)/(Y3-YC): LET Y
3=90
2420 IF Y4<2 THEN  LET X4=128-(
128-X4)*(2-YC)/(Y4-YC): LET Y4=
2: GOTO 2440
2430 IF Y4>90 THEN  LET X4=128-
(128-X4)*(90-YC)/(Y4-YC): LET Y
4=90
2440 IF X3<11 OR X4>244 THEN  G
OTO 2500
2445 OVER 1: PLOT X3,176-Y3: DR
AW X4-PEEK 23677,(176-Y4)-PEEK
23678: OVER 0
2500 GOSUB 8000
2505 IF GC=0 THEN  LET GC=1
2510 LET RR=RL: LET PP=PT: RETU
RN
5080 LET GZ=VZ: LET GY=VY+WY: L
ET GX=VX+WX
5090 LET TC=5
5100 LET RT=3: LET TP=5: LET WR
=50
5500 IF INT (RND*5)=1 THEN  LET
 RL=RL+INT (RND*5)-2: IF INT (R
ND*5)=1 THEN  LET PT=PT+3-INT (
RND*2)+1*2
5510 GOSUB 1000: IF PZ<0 THEN
GOTO 5530
5520 GOSUB 2180: GOTO 5500
5530 GOTO 5500
8000 IF GC<>0 THEN  PLOT 127,17
4: DRAW  OVER 1;OX,OY
8010 LET OX=16*SIN (RB*(PI/180)
): LET OY=-(16*ABS COS (RB*(PI/
180)))
8020 PLOT 127,174: DRAW  OVER 1
;OX,OY
8025 LET WB=AD: IF AD>180 THEN
 LET WB=WB-360
8026 IF RB>180 THEN  LET WB=WB+
360-RB: GOTO 8040
8030 LET WB=WB-RB
8040 IF RW=1 THEN  PLOT  OVER 1
;RDX,175-RDY
8050 LET RW=0: IF ABS WB>57 THE
N  RETURN
8060 LET RDX=X3+INT (((X4-X3)/2
)-SIN (WB*(PI/180))*(X4-X3)*6)
8070 LET RDY=Y3+((Y4-Y3)*((RDX-
X3)/(X4-X3))+2)
8080 IF RDY<2 OR RDY>90 OR RDX<
11 OR RDX>244 THEN  RETURN
```



```
8090 LET RW=1: PLOT  OVER 1;RDX
,175-RDY
8100 RETURN
```

O **POKE** na linha 1 trava as linhas maiúsculas do computador. A linha 5 zera todas as variáveis. A linha 110, que já foi digitada na primeira parte do programa (ou seja, no artigo anterior), envia o computador para a linha 5000, onde é desenhado o interior da cabine. A linha 5080 cuida das variáveis que determinam a posição do avião no espaço: **GZ** se refere à distância do aparelho ao longo do eixo **Z** — sua altitude; **VZ** é a velocidade ao longo desse mesmo eixo; **VY** é a velocidade ao longo do eixo **Y** — para frente e para trás; **WY** é a velocidade do vento nessa mesma direção. **GX**, **VX** e **WX** correspondem à distância percorrida, velocidade e velocidade do vento em relação ao eixo X — direita e esquerda.

A linha 5090 estabelece o valor inicial da rotação do motor. A linha 5100 define os limites da inclinação do avião (**RT**, relacionada com os movimentos laterais), assim como da ascensão (**TP**, que faz o avião subir ou descer) e da largura da pista (**WR**).

A linha 5500 é um comando temporário encarregado de manobrar o avião até o momento em que a rotina de controle propriamente dita é digitada (o que será feito no próximo artigo desta série).

A linha 5510 chama a sub-rotina que começa na linha 1000 e termina em 1110. Essa sub-rotina recalcula os valores de todas as variáveis à medida que o avião se movimenta. As linhas 1025 e 1030 verificam se o aparelho entrou em parafuso, por ter uma velocidade muito baixa (inferior a 30 m por segundo). Caso isso aconteça, será chamada a sub-rotina 1500, que se encarrega de reproduzir os efeitos de um mergulho. A linha 5510 impede que possamos prever o rumo do avião durante a queda (este pode mergulhar diretamente ou então entrar em parafuso).

A linha 1100 chama a sub-rotina 500, que calcula o ângulo da trajetória do avião.

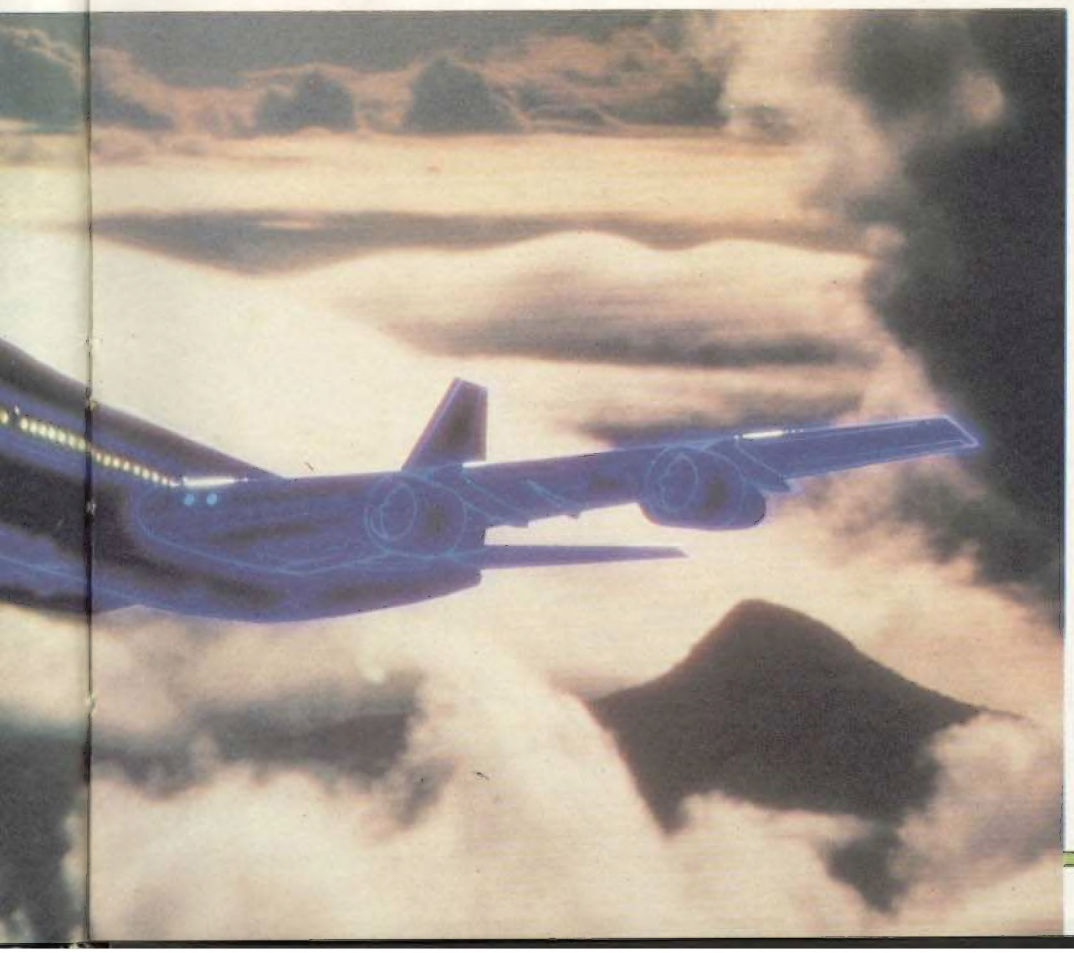
A próxima grande sub-rotina a ser chamada começa na linha 2180 e termina em 2510. Ela redesenha os mostradores e contadores à medida que a informação que estes trazem se modifica. A linha 2180 verifica o contador **GC** para ver se há algo a ser redesenhado. A sub-rotina 2200 desenha a nova posição dos ponteiros do mostrador da velocidade do vento (*airspeed*). As linhas 2205 e 2210 calculam e redesenham as duas novas posições dos ponteiros que marcam a altitude. A linha 2230 movimenta os ponteiros, usando os cálculos das linhas 2225 e 2226.

O contagiros é atualizado pela linha 2240. As linhas 2250 a 2270 calculam a direção da pista (*runway*) e o desvio do avião em relação ao eixo principal desta última (*drift*). A distância até o centro da pista é calculada e impressa pela linha 2280.

O horizonte artificial do segundo mostrador é calculado e desenhado pelas linhas 2280 a 2370.

A linha 2376 verifica o horizonte que aparece na janela, enquanto as linhas 2377 a 2445 o recalculam e o redesenham, conforme o caso.

A linha 2500 chama a sub-rotina que começa na linha 8000 e termina na 8100. Essa sub-rotina calcula e desenha a imagem da pista no radar, que pode ser vista no topo da tela, e o pontinho logo abaixo da linha de horizonte.



```
100 PRINT RND(-TIME)
500 RA=AD*C:VX=VV*SIN(RA)
510 VY=VV*COS(RA):RETURN
1000 PZ=PZ+GZ:PY=PY+GY:PX=PX+GX
1020 VV=VV+16*(TC*30-VV-8*PT)/V
V
1030 IF ST=1 THEN LINE(90,1)-(2
30,8),4,BF:ST=0:GOTO 1050
1040 IF VV<30 THEN GOSUB 1500
1050 AD=AD+RL:IF AD<0 THEN AD=A
D+360
1060 IF AD>359 THEN AD=AD-360
1070 VZ=VV*SIN(PT*C)-10+VV/15
1080 GZ=VZ:GY=VY+WY:GX=VX+WX
1090 IF VY=0 THEN GD=-PI/2:GOTO
 1110
1100 GD=-ATN(VX/VY)/C
1110 GOSUB 500
1120 RETURN
1500 ST=1:X=96:Y=1:COLOR 6
1510 A$="MERGULHO":GOSUB 4000:X
=97:GOSUB 4000
1520 PLAY"T255AGFEDCAGFEDC"
1530 RL=INT(RND(1)*21)-9:PT=-21
-INT(RND(1)*5)
1540 RETURN
2000 COLOR 15
2005 LINE(35,128)-(35+15*SIN(V1
*PI/200),128-15*COS(V1*PI/200))
,8
2010 LINE(35,128)-(35+15*SIN(VV
*PI/200),128-15*COS(VV*PI/200))
2015 LINE(155,128)-(155+10*SIN(
T1*PI/5),128-10*COS(T1*PI/5)),8
2020 TN=PZ/1000:UN=PZ-1000*INT(
TN):LINE(155,128)-(155+10*SIN(T
N*PI/5),128-10*COS(TN*PI/5))
2025 LINE(155,128)-(155+15*SIN(
U1*PI/500),128-15*COS(U1*PI/500
)),8
2030 LINE(155,128)-(155+15*SIN(
UN*PI/500),128-15*COS(UN*PI/500
))
2035 LINE(215,128)-(215+15*SIN(
T2*PI/5),128-15*COS(T2*PI/5)),8
2040 LINE(215,128)-(215+15*SIN(
TC*PI/5),128-15*COS(TC*PI/5))
2050 COLOR 15:X=16:Y=180:A$=STR
```

```
$(ABS(INT(AD))):GOSUB 3900
2060 IF PY=0 THEN RB=0 ELSE RB=
ATN(PX/PY)/C:IF PY>0 THEN RB=RB
+180
2070 IF RB<0 THEN RB=RB+360
2080 X=79:A$=STR$(INT(RB)):GOSU
B 3900:X=138:A$=STR$(INT(PX)):G
OSUB 3900
2090 X=194:A$=STR$(INT(SQR(PY*P
Y+PX*PX))):GOSUB 3900
2100 YC=128+PT:X1=80:X2=110:Y1=
YC+17*TAN(RL*2*C):Y2=YC-17*TAN(
RL*2*C)
2110 IF(YC<113 OR YC>143) AND R
L=0 THEN 2320
2120 IF Y1<113 THEN X1=95-(95-X
1)*(105-YC)/(Y1-YC):Y1=113:GOTO
 2140
2130 IF Y1>143 THEN X1=95-(95-X
1)*(135-YC)/(Y1-YC):Y1=143
2140 IF Y2<113 THEN X2=95-(95-X
2)*(105-YC)/(Y2-YC):Y2=113:GOTO
 2160
2150 IF Y2>143 THEN X2=95-(95-X
2)*(135-YC)/(Y2-YC):Y2=143
2160 IF X1<80 OR X2>110 THEN 21
80
2165 DRAW "BM81,128C10R9F5E5R9"
2170 LINE(X5,Y5)-(X6,Y6),8:LINE
(X1,Y1)-(X2,Y2):X5=X1:Y5=Y1:X6=
X2:Y6=Y2
2180 IF X7-R>10 AND X7+R<245 AN
D Y7>0 AND Y7<80 THEN CIRCLE(X7
,Y7),R,4,0,.5,10
2190 IF RL=RR AND PP=PT THEN 22
90
2200 IF HF=1 THEN LINE (X3,Y3)-
(X4,Y4),4
2210 HF=0:YC=33+PT*4:X3=16:X4=2
39:Y3=YC+118*TAN(RL*2*C):Y4=YC-
118*TAN(RL*2*C)
2220 IF(YC<10 OR YC>79) AND RL=
0 THEN 2290
2230 IF Y3<1 THEN X3=128-(128-X
3)*(1-YC)/(Y3-YC):Y3=1:GOTO 225
0
2240 IF Y3>79 THEN X3=128-(128-
X3)*(79-YC)/(Y3-YC):Y3=79
2250 IF Y4<1 THEN X4=128-(128-X
4)*(1-YC)/(Y4-YC):Y4=1:GOTO 227
0
2260 IF Y4>79 THEN X4=128-(128-
X4)*(79-YC)/(Y4-YC):Y4=79
2270 IF X3<16 OR X4>239 THEN 22
90
2280 HF=1:LINE(X3,Y3)-(X4,Y4),1
5
2290 WB=AD:IF AD>180 THEN WB=WB
-360
2300 IF RB>180 THEN WB=WB+360-R
B ELSE WB=WB-RB
2310 IF ABS(WB)>60 AND ABS(PY)>
1000 THEN 2370
2320 AN=118/(60*SQR((X3-X4)*(X3
-X4)+(Y3-Y4)*(Y3-Y4))):X7=(X3+X
4)/2+SGN(X3-X4)+WB*AN*(X3-X4)
2330 Y7=(Y3+Y4)/2+2+WB*AN*(Y3-Y
4):IF X7<16 OR X7>239 OR Y7<1 O
R Y7>79 THEN 2370
2340 IF ABS(PY)<1000 THEN R=8-Y
7/10 ELSE R=4000/ABS(PY):IF R*1
0+Y7>80 THEN R=8-Y7/10
2350 IF Y7<1 OR Y7>79 OR X7-R<1
6 OR X7+R>239 THEN 2370
2360 CIRCLE(X7,Y7),R,15,0,.5,10
2370 V1=VV:U1=UN:T1=TN:T2=TC
2380 RR=RL:PP=PT:RETURN
3900 LINE(X,Y)-(X+50,Y+8),2,BF
5080 GZ=VZ:GY=VY+WY:GX=VX+WX
5090 TC=5
5100 RT=3:TP=5:WR=50
5190 DRAW "BM81,128C10R9F5E5R9"
5500 IF INT(RND(1)*10)=1 THEN R
L=RL-SGN(RL)+INT(RND(1)*5-2)
5505 IF INT(RND(1)*10)=1 THEN P
T=PT+3-INT(RND(1)*2+1)*2
5510 GOSUB 1000:IF PZ<0 THEN 55
30
5520 GOSUB 2000:GOTO 5500
5530 GOTO 5530
```

A linha 100 "embaralha as cartas", permitindo a criação de números ao acaso. A 5080 estabelece os valores das variáveis que controlam a posição do aparelho no espaço, conforme a sua velocidade e a força e direção do vento. **GZ** nos revela a distância percorrida ao longo do eixo Z (alto-baixo); **VZ**, a velocidade nesse eixo. **VY** informa a velocidade ao longo do eixo **Y** (frente-trás); **WY**, a velocidade do vento nessa mesma direção; **GX**, **VX** e **WX** correspondem respectivamente à distância percorrida, à velocidade do aparelho e à velocidade do vento ao longo do eixo **X** (direita-esquerda).

A linha 5090 determina a rotação inicial do motor; a 5100 define os limites da inclinação lateral (**RT**), que faz o avião virar; da inclinação frontal (**TP**), que faz o avião subir ou descer, e da largura da pista (**WR**).

As últimas cinco linhas são comandos temporários. Elas fazem com que o avião voe de maneira irregular, com inclinações laterais e frontais estabelecidas ao acaso.

As sub-rotinas das linhas 1000 a 1120 recalculam todas as variáveis enquanto o aeroplano se movimenta. As linhas 1030 e 1040 verificam se a velocidade caiu abaixo de 30 m por segundo. Quando isso acontece, é chamada a sub-rotina da linha 1500, que recria os efeitos de uma queda ou mergulho. A linha 1510 introduz um elemento de acaso no processo, de maneira que tanto podemos mergulhar diretamente para o chão como podemos entrar em parafuso. A sub-rotina 500, que é chamada pela linha 1110, calcula o ângulo em que o aparelho está voando.

A sub-rotina que vai da linha 2000 à linha 2380 cuida do restante do programa. Ela atualiza a tela, ajustando o painel de instrumentos, que mostra o movimento do avião. Para fazer isso, é necessário, primeiramente, apagar o antigo conteúdo de cada instrumento e desenhar a seguir os ponteiros correspondentes à nova posição do aparelho.

O mostrador da velocidade do vento é atualizado pelas linhas 2005 e 2010, enquanto as linhas 2015, 2020, 2025 e 2030 cuidam do mostrador de altitude. O contagiros é modificado pelas linhas 2035 e 2040; a linha 2050 corrige a direção em que o avião está seguindo (*bearing*, que é o ângulo da bússola medido em graus). A direção da pista (*runway*, também em graus) é modificada pelas linhas 2060 a 2080, e o novo valor da distância é impresso pela linha 2090.

A linha 2190 verifica se o horizonte deve ser modificado, enquanto as linhas 2100 a 2170 desenham o novo horizonte artificial no mostrador correspondente. O horizonte é desenhado pelas linhas 2200 a 2280.

A nova posição da pista é calculada pelas linhas 2290 a 2360. A 2350 verifica se a pista pode ser desenhada na tela. A 2180 apaga a versão antiga quando uma nova é desenhada.

Execute então o programa e veja o avião voar sozinho... por enquanto.

```
800 RA = AD * C:VX = AS *  SIN
(RA)
810 VY = AS *  COS (RA): RETURN

1000 PZ = PZ + GZ:PY = PY + GY:
PX = PX + GX
1020 AS = AS + 16 * (TC * 30 -
AS - 8 * PT) / AS
1030  IF ST = 1 THEN ST = 0: GO
TO 1050
1040  IF AS < 30 THEN  GOSUB 15
00
1050 AD = AD + RL: IF AD < 0 TH
EN AD = AD + 360
1060  IF AD > 359 THEN AD = AD
- 360
1070 VZ = AS *  SIN (PT * C) -
10 + AS / 15
1080 GZ = VZ:GY = VY + WY:GX =
VX + WX
1090  IF VY = 0 THEN GD =  - PI
 / 2: GOTO 1110
1100 GD =  -  ATN (VX / VY) / C
1110  GOSUB 800
1120  RETURN
1500 ST = 1
1510 RL =  INT ( RND (1) * 21 +
 1) - 10:PT =  - 20 -  INT ( RN
D (1) * 5 + 1)
1520  RETURN
2000  HCOLOR= 0
2010  HPLOT 41,134 TO 41 + 20 *
  SIN (A1 * PI / 200),134 - 20
*  COS (A1 * PI / 200)
2015  HCOLOR= 3: HPLOT 41,134 T
O 41 + 20 *  SIN (AS * PI / 200
),134 - 20 *  COS (AS * PI / 20
0)
2020  HCOLOR= 0: HPLOT 177,134
TO 177 + 15 *  SIN (T1 * PI / 5
```

```
),134 - 15 *  COS (T1 * PI / 5)
2025  HCOLOR= 3:TN = PZ / 1000:
UN = PZ - 1000 *  INT (TN): HPL
OT 177,134 TO 177 + 15 *  SIN (
TN * PI / 5),134 - 15 *  COS (T
N * PI / 5)
2030  HCOLOR= 0: HPLOT 177,134
TO 177 + 20 *  SIN (U1 * PI / 5
00),134 - 20 *  COS (U1 * PI /
500)
2035  HCOLOR= 3: HPLOT 177,134
TO 177 + 20 *  SIN (UN * PI / 5
00),134 - 20 *  COS (UN * PI /
500)
2040  HCOLOR= 0: HPLOT 245,134
TO 245 + 20 *  SIN (T2 * PI / 5
),134 - 20 *  COS (T2 * PI / 5)
2045  HCOLOR= 3: HPLOT 245,134
TO 245 + 20 *  SIN (TC * PI / 5
),134 - 20 *  COS (TC * PI / 5)
2050 L = 23:C = 4:A$ =  LEFT$ (
 STR$ ( ABS ( INT (AD))) + "
",3): GOSUB 4000
2060  IF PY = 0 THEN RB = 0: GO
TO 2070
2062 RB =  ATN (PX / PY) / C
2065  IF PY > 0 THEN RB = RB +
180
2070  IF RB < 0 THEN RB = RB +
360
2080 C = 14:A$ =  LEFT$ ( STR$
( INT (RB)) + "   ",3): GOSUB 4
000:C = 24:A$ =  LEFT$ ( STR$ (
 INT (PX)) + "    ",4): GOSUB 4
000
2090 C = 32:A$ =  LEFT$ ( STR$
( INT ( SQR (PY * PY + PX * PX)
)) + "     ",5): GOSUB 4000
2100 YC = 134 + PT:X1 = 90:X2 =
 130:Y1 = YC + 17 *  TAN (RL *
2 * C):Y2 = YC - 17 *  TAN (RL
* 2 * C)
2110  IF (YC < 119 OR YC > 149)
 AND RL = 0 THEN 2380
2120  IF Y1 < 119 THEN X1 = 110
 - (110 - X1) * (119 - YC) / (Y
1 - YC):Y1 = 119: GOTO 2140
2130  IF Y1 > 149 THEN X1 = 110
 - (110 - X1) * (149 - YC) / (Y
1 - YC):Y1 = 149
2140  IF Y2 < 119 THEN X2 = 110
 - (110 - X2) * (119 - YC) / (Y
2 - YC):Y2 = 119
2150  IF Y2 > 149 THEN X2 = 110
 - (110 - X2) * (149 - YC) / (Y
2 - YC):Y2 = 149
2160  IF X1 < 90 OR X2 > 130 TH
EN 2190
2170  HCOLOR= 0: HPLOT X5,Y5 TO
 X6,Y6
2175  HCOLOR= 3: HPLOT X1,Y1 TO
 X2,Y2
2177  HCOLOR= 5: HPLOT 85,138 T
O 104,138 TO 109,148 TO 114,138
 TO 134,138
2190  IF RL = RR AND PP = PT TH
EN 2380
2200  IF HF = 1 THEN  HCOLOR= 0
: HPLOT X3,Y3 TO X4,Y4
2210 HF = 0:YC = 33 + PT * 4:X3
 = 11:X4 = 275:Y3 = YC + 118 *
 TAN (RL * 2 * C):Y4 = YC - 118
 *  TAN (RL * 2 * C)
2220  IF (YC < 10 OR YC > 79) A
ND RL = 0 THEN 2380
2230  IF Y3 < 1 THEN X3 = 143 -
 (143 - X3) * (1 - YC) / (Y3 -
YC):Y3 = 1: GOTO 2250
2240  IF Y3 > 79 THEN X3 = 143
- (143 - X3) * (79 - YC) / (Y3
- YC):Y3 = 79
2250  IF Y4 < 1 THEN X4 = 143 -
 (143 - X4) * (1 - YC) / (Y4 -
YC):Y4 = 1: GOTO 2270
2260  IF Y4 > 79 THEN X4 = 143
- (143 - X4) * (79 - YC) / (Y4
- YC):Y4 = 79
2270  IF X3 < 11 OR X4 > 275 TH
EN 2380
2280 HF = 1: HCOLOR= 3: HPLOT X
3,Y3 TO X4,Y4
2380 RR = RL:PP = PT:A1 = AS:U1
 = UN:T1 = TN:T2 = TC:X5 = X1:Y
5 = Y1:X6 = X2:Y6 = Y2: RETURN
5080 GZ = VZ:GY = VY + WY:GX =
VX + WX
5090 TC = 5
5100 RT = 3:TP = 5:WR = 50
5500  IF  INT ( RND (1) * 10 +
1) = 1 THEN RL = RL -  SGN (RL)
 + ( INT ( RND (1) * 5 + 1) - 3
)
5505  IF  INT ( RND (1) * 10 +
1) = 1 THEN PT = PT + 3 -  INT
( RND (1) * 2 + 1) * 2
5510  GOSUB 1000: IF PZ < 0 THE
N 5530
5520  GOSUB 2000: GOTO 5500
5530  GOTO 5530
```

A linha 5080 estabelece os valores das variáveis que controlam a posição do aparelho no espaço, de acordo com a sua velocidade e a força e a direção do vento. **GZ** informa sobre a distância percorrida ao longo do eixo **Z** (alto-baixo); **VZ** nos dá a velocidade nesse eixo; **VY**, a velocidade ao longo do eixo **Y** (frente-trás); **WY**, a velocidade do vento nessa mesma direção. Da mesma forma, **GX**, **VX** e **WX** correspondem, respectivamente, à distância percorrida, à velocidade do avião e à velocidade do vento ao longo do eixo **X** (direita-esquerda).

A linha 5090 determina a rotação inicial do motor, a 5100 define os limites da inclinação lateral (**RT**), que faz o avião virar; da inclinação frontal (**TP**), que faz o avião subir ou descer, e da largura da pista (**WR**).

As últimas cinco linhas são comandos temporários. Elas fazem com que o avião voe de maneira irregular, com inclinações laterais e frontais estabelecidas ao acaso.

As sub-rotinas das linhas 1000 a 1120 recalculam todas as variáveis enquanto o aeroplano se movimenta. As linhas 1030 e 1040 verificam se a velocidade caiu abaixo de 30 m por segundo. Quando isso acontece, é chamada a sub-rotina da linha 1500, que recria os efeitos de uma queda ou mergulho. A linha 1510 introduz um elemento de acaso no processo, de forma que tanto podemos mergulhar diretamente para o chão como entrar em parafuso. A sub-rotina 500, que é chamada pela linha 1110, é encarregada de calcular o ângulo em que o aparelho está voando.

A sub-rotina que vai da linha 2000 à 2380 cuida do restante do programa. Ela atualiza a tela, ajustando o painel de instrumentos, que mostra o movimento do avião. Para isso, é necessário apagar o antigo conteúdo de cada instrumento e desenhar os ponteiros correspondentes à nova posição.

O mostrador da velocidade do vento é atualizado pelas linhas 2010 e 2015, enquanto as linhas 2020, 2025, 2030 e 2035 cuidam do mostrador de altitude. O contagiros, por sua vez, é modificado pelas linhas 2040 e 2045; a linha 2050 corrige a direção em que o avião está voando (*bearing*, que é o ângulo da bússola medido em graus). A direção da pista (*runway*, também em graus) é modificada pelas linhas 2060 a 2080, e o novo valor da distância é impresso pela linha 2090.

A linha 2090 também verifica se o horizonte deve ser modificado, e as linhas 2100 a 2170 desenham o novo horizonte artificial no mostrador correspondente. O horizonte verdadeiro é desenhado pelas linhas 2200 a 2280.

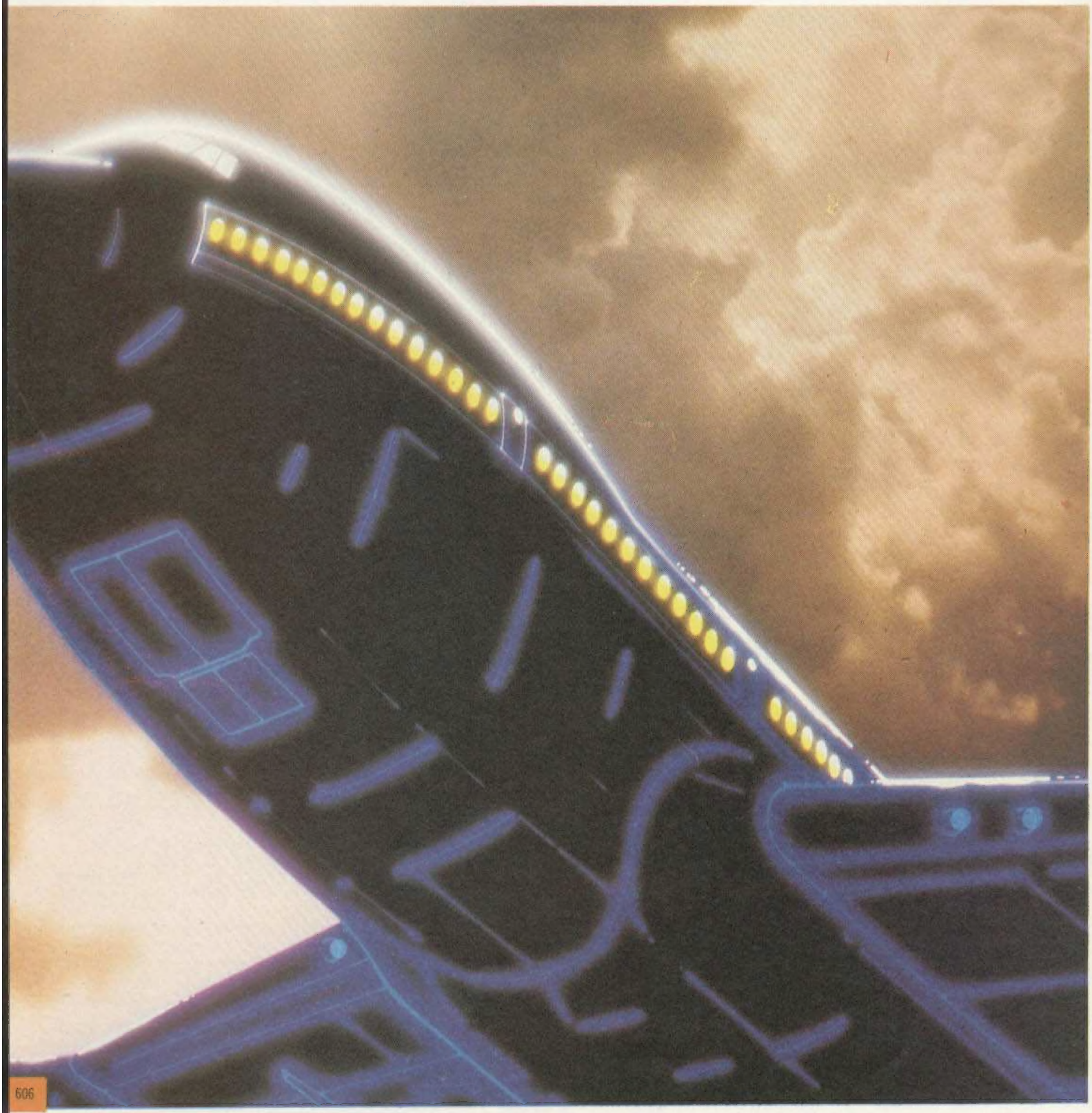
Execute então o programa e veja o avião voar sob o controle do piloto automático.

É possível que, após alguns minutos de vôo, a tela gráfica comece a ser preenchida por pontos aleatórios que estragam o desenho. Isso acontece sempre que usamos muitos cordões. No nosso caso, é a variável A$ (responsável pela impressão dos contadores) que é usada a todo instante.

Não se alarme. O próximo artigo resolverá o problema, e permitirá que você assuma o controle do aparelho.

```
500 RA=AD*C:VX=VV*SIN(RA)
510 VY=VV*COS(RA):RETURN
1000 PZ=PZ+GZ:PY=PY+GY:PX=PX+GX
1020 VV=VV+16*(TC*30-VV-8*PT)/V
V
1030 IF ST=1 THEN PMODE 4,1:DRA
W"BM108,40C0":A$="MERGULHO":GOS
UB 4000:DRAW"C5":ST=0:GOTO 1050
1040 IF VV<30 GOSUB 1500
1050 AD=AD+RL:IF AD<0 THEN AD=A
D+360
1060 IF AD>359 THEN AD=AD-360
1070 VZ=VV*SIN(PT*C)-10+VV/15
1080 GZ=VZ:GY=VY+WY:GX=VX+WX
1090 IF VY=0 THEN GD=-PI/2:GOTO
 1110
```

```
1100 GD=-ATN(VX/VY)/C
1110 GOSUB 500
1120 RETURN
1500 PMODE 4,1:ST=1:DRAW"BM108,
40":A$="MERGULHO":GOSUB 4000:PL
AY"T20AGFEDCAGFEDC"
1510 RL=RND(21)-10:PT=-20-RND(5
)
1520 RETURN
2000 PCOPY 5 TO 7:PCOPY 6 TO 8:
PMODE 4,5
2010 LINE(35,120)-(35+20*SIN(VV
*PI/200),120-20*COS(VV*PI/200))
,PSET
2020 TN=PZ/1000:UN=PZ-1000*INT(
TN):LINE(155,120)-(155+15*SIN(T
N*PI/5),120-15*COS(TN*PI/5)),PS
ET
2030 LINE(155,120)-(155+20*SIN(
UN*PI/500),120-20*COS(UN*PI/500
)),PSET
2040 LINE(215,120)-(215+20*SIN(
TC*PI/5),120-20*COS(TC*PI/5)),P
SET
2050 DRAW"BM16,172S8":A$=STR$(A
BS(INT(AD))):GOSUB 4000
2060 IF PY=0 THEN RB=0 ELSE RB=
```



```
ATN(PX/PY)/C:IF PY>0 THEN RB=RB
+180
2070 IF RB<0 THEN RB=RB+360
2080 DRAW"BM80,172":A$=STR$(INT
(RB)):GOSUB 4000:DRAW"BM140,172
":A$=STR$(ABS(INT(PX))):GOSUB 4
000
2090 DRAW"BM196,172":A$=STR$(IN
T(SQR(PY*PY+PX*PX))):GOSUB 4000
2100 YC=120+PT:X1=80:X2=110:Y1=
YC+17*TAN(RL*2*C):Y2=YC-17*TAN(
RL*2*C)
2110 IF (YC<105 OR YC>135)AND R
L=0 THEN 2320
2120 IF Y1<105 THEN X1=95-(95-X
1)*(105-YC)/(Y1-YC):Y1=105:GOTO
 2140
2130 IF Y1>135 THEN X1=95-(95-X
1)*(135-YC)/(Y1-YC):Y1=135
2140 IF Y2<105 THEN X2=95-(95-X
2)*(105-YC)/(Y2-YC):Y2=105:GOTO
 2160
2150 IF Y2>135 THEN X2=95-(95-X
2)*(135-YC)/(Y2-YC):Y2=135
2160 IF X1<80 OR X2>110 THEN 21
80
2170 LINE(X1,Y1)-(X2,Y2),PSET
2180 PMODE 4,1:IF X5-R>10 AND X
5+R<245 AND Y5>0 AND Y5<80 THEN
 CIRCLE (X5,Y5),R,0,10,0,.5
2190 IF RL=RR AND PP=PT THEN 22
90
2200 IF HF=1 THEN LINE (X3,Y3)-
(X4,Y4),PRESET
2210 HF=0:YC=33+PT*4:X3=11:X4=2
44:Y3=YC+118*TAN(RL*2*C):Y4=YC-
118*TAN(RL*2*C)
2220 IF(YC<10 OR YC>79)AND RL=0
 THEN 2290
2230 IF Y3<1 THEN X3=128-(128-X
3)*(1-YC)/(Y3-YC):Y3=1:GOTO 225
0
2240 IF Y3>79 THEN X3=128-(128-
X3)*(79-YC)/(Y3-YC):Y3=79
2250 IF Y4<1 THEN X4=128-(128-X
4)*(1-YC)/(Y4-YC):Y4=1:GOTO 227
0
2260 IF Y4>79 THEN X4=128-(128-
X4)*(79-YC)/(Y4-YC):Y4=79
2270 IF X3<11 OR X4>244 THEN 22
90
2280 HF=1:LINE(X3,Y3)-(X4,Y4),P
SET
2290 WB=AD:IF AD>180 THEN WB=WB
-360
2300 IF RB>180 THEN WB=WB+360-R
B ELSE WB=WB-RB
2310 IF ABS(WB)>60 AND ABS(PY)>
1000 THEN 2370
2320 AN=118/(60*SQR((X3-X4)*(X3
-X4)+(Y3-Y4)*(Y3-Y4))):X5=(X3+X
4)/2+SGN(X3-X4)+WB*AN*(X3-X4)
2330 Y5=(Y3+Y4)/2+2+WB*AN*(Y3-Y
4):IF X5<11 OR X5>244 OR Y5<1 O
R Y5>79 THEN 2370
2340 IF ABS(OY)<1000 THEN R=8-Y
5/10 ELSE R=4000/ABS(PY):IF R*1
0+Y5>80 THEN R=8-Y5/10
2350 IF Y5<10 OR Y5>79 OR X5-R<
11 OR X5+R>244 THEN 2370
2360 CIRCLE (X5,Y5),R,5,10,0,.5
2370 PCOPY 7 TO 3:PCOPY 8 TO 4
2380 RR=RL:PP=PT:RETURN
5080 GZ=VZ:GY=VY+WY:GX=VX+WX
5090 TC=5
5100 RT=3:TP=5:WR=50
5500 IF RND(10)=1 THEN RL=RL-SG
N(RL)+(RND(5)-3)
5505 IF RND(10)=1 THEN PT=PT+3-
RND(2)*2
5510 GOSUB 1000:IF PZ<0 THEN 55
30
5520 GOSUB 2000:GOTO 5500
5530 GOTO 5530
```

A linha 5080 estabelece os valores das variáveis que controlam a posição do aparelho no espaço, conforme a sua velocidade e a força e direção do vento. **GZ** revela a distância percorrida ao longo do eixo **Z** (alto-baixo); **VZ**, a velocidade neste eixo; **VY**, a velocidade ao longo do eixo **Y** (frente-trás); **WY**, a velocidade do vento nessa mesma direção. Por sua vez, **GX**, **VX** e **WX** correspondem, respectivamente, à distância percorrida, à velocidade do aparelho e à velocidade do vento ao longo do eixo **X** (direita-esquerda).

A linha 5090 determina a rotação inicial do motor; a 5100 define os limites da inclinação lateral (**RT**), que faz o avião virar; da inclinação frontal (**TP**), que faz o avião subir ou descer, e da largura da pista (**WR**).

As últimas cinco linhas são comandos temporários. Elas fazem com que o avião voe de maneira irregular, com inclinações laterais e frontais estabelecidas ao acaso.

A sub-rotina das linhas 1000 a 1120 recalcula todas as variáveis enquanto o aeroplano se movimenta. As linhas 1030 e 1040 verificam se a velocidade caiu abaixo de 30 m por segundo. Quando isso acontece, é chamada a sub-rotina da linha 1500, que recria os efeitos de uma queda ou mergulho. A linha 1510 introduz um elemento de acaso no processo, de forma que tanto podemos mergulhar diretamente para o chão como entrar em parafuso. A sub-rotina 500, que é chamada pela linha 1110, é incumbida de calcular o ângulo em que o aparelho está voando.

A sub-rotina que vai da linha 2000 a 2380 cuida do restante do programa. Ela atualiza a tela, ajustando o painel de instrumentos que mostram o movimento do avião.

A linha 2000 transfere a versão original da cabine para outra parte da memória. O mostrador da velocidade do vento é atualizado pela linha 2010, enquanto as linhas 2020 e 2030 cuidam do mostrador de altitude. O contagiros é modificado pela linha 2040, enquanto a 2050 corrige a direção em que o avião está seguindo (*bearing*, que é o ângulo da bússola medido em graus). A direção da pista (*runway*, também em graus) é modificada pelas linhas 2060 a 2080, e o novo valor da distância é impresso pela linha 2090.

A linha 2190 verifica se o horizonte deve ser modificado, enquanto as linhas 2100 a 2170 desenham o novo horizonte artificial no mostrador correspondente. O horizonte verdadeiro é desenhado pelas linhas 2200 a 2280.

A nova posição da pista é calculada pelas linhas 2290 a 2360. A linha 2350 verifica se a pista pode ser desenhada na tela (a pista aumenta de tamanho à medida que o avião se aproxima). A linha 2180 apaga a versão antiga quando uma nova é desenhada.

Todas essas modificações foram feitas de maneira invisível, em uma página oculta da tela. A linha 2370 transfere o desenho para a tela usando **PCOPY**, de modo que todas as alterações parecem ocorrer simultaneamente.

Como melhorar a velocidade de execução do simulador de vôo?

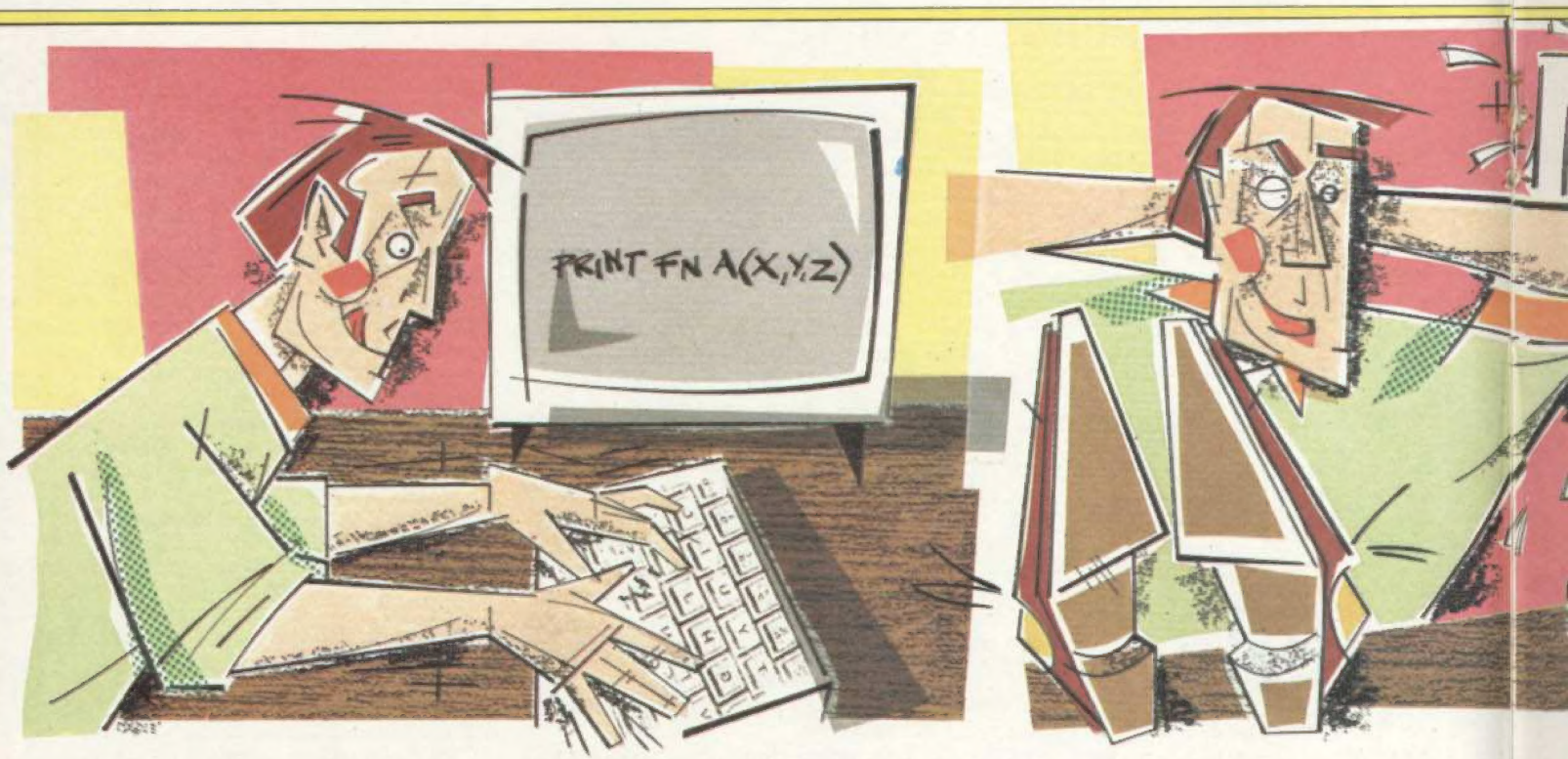
O programa de simulação de vôo é bastante lento. Isso acontece porque ele foi escrito em linguagem BASIC interpretada, com um grande número de equações a serem resolvidas a cada passo da simulação, e de gráficos complexos a serem desenhados. Uma das soluções para o problema seria colocar mais comandos por linha, e tirar espaços em branco; isso, porém, tornaria o programa mais difícil de ser entendido e corrigido, sem aumentar muito a velocidade.

Uma saída (para quem dispõe de um micro com unidade acionadora de disquetes) consiste em utilizar um *compilador* para o programa. Um compilador é um programa especial que traduz um outro programa desenvolvido em linguagem de alto nível (como o BASIC) para código de máquina. Com ele, a velocidade de execução chega a aumentar de dez a trinta vezes, dependendo do modelo usado.

Quase todos os micros dispõem de compiladores para a linguagem BASIC. É o caso das linhas Apple, TRS-80 e TRS-Color, por exemplo. O Spectrum e o ZX-81, porém, não contam com esse tipo de recurso. Para a linha MSX, o compilador BASIC é encontrado apenas no exterior.

FUNÇÕES SOB ENCOMENDA

Calcular juros compostos? Encontrar o cubo de um número? Transformar letras maiúsculas em minúsculas? Tudo isso se torna simples quando se trabalha com funções sob encomenda.



Um computador é capaz de resolver problemas matemáticos muito mais rapidamente do que qualquer pessoa. Entretanto, ele só faz o que for programado para fazer, seja por você, seja por um programa pronto, carregado previamente na máquina.

A linguagem mais utilizada para programação de microcomputadores é o BASIC. Em sua maioria, os micros pessoais passam a aceitar comandos nessa linguagem assim que são ligados.

O interpretador BASIC de seu computador conta com uma série de funções matemáticas padronizadas: com elas, o programador pode solicitar cálculos matemáticos complexos, como raiz quadrada, logaritmo, funções trigonométricas e outros. Cada uma dessas funções é definida de tal modo que, assim que reconhece a função, o interpretador toma o valor fornecido para ela pelo programador e executa a operação apropriada. O nome dado às funções evoca as operações que elas realizam: por exemplo, **SQR** (*square root*, ou raiz quadrada, em inglês), **LOG** (logaritmo), **COS** (cosseno) etc.

FUNÇÕES BÁSICAS

Tudo corre bem quando as funções que se pretende usar em um programa existem no BASIC do computador. Muitas funções matemáticas e não matemáticas, porém, não fazem parte do BASIC padrão. Por exemplo, não há nenhuma função chamada **CUBE**, que seria muito útil para calcular automaticamente o cubo de um número (n x n x n).

Mas nem tudo está perdido: felizmente há alternativas para quando o programador precisar de uma função não disponível no BASIC. A primeira seria evitar o emprego de funções no programa: cada vez que um cálculo de cubo precisa ser realizado, por exemplo, pode-se colocar a expressão completa dentro de um comando **LET**. Se isso não for possível — pois o programa ficaria desnecessariamente longo —, pode-se programar uma sub-rotina, que efetuará os cálculos necessários.

A solução mais elegante, entretanto, consiste em um recurso de muitos interpretadores BASIC, que permite a definição de *funções do usuário*. Estas agem como funções de encomenda, que atuam apenas dentro do programa no qual foram definidas. Essa opção existe para os usuários de computadores das linhas Spectrum, TRS-80 (com BASIC nível III), TRS-Color, MSX, Apple II e TK-2000. Os micros das linhas ZX-81 e TRS-80 (baseado em fita cassete) não oferecem essa possibilidade. Em alguns micros, o recurso de definição de funções é bastante poderoso; em outros, como veremos, é mais limitado.

As funções definidas pelo usuário dão ao programador a capacidade de ajustar o BASIC do computador conforme as necessidades de um programa em particular. Os comandos usados para definir uma função diferem ligeiramente de computador para computador, mas seguem em geral esta forma básica:

```
DEF FN a (x) = expressão
```

Aqui, a letra **a** é o nome da função (necessário para quando você quiser utilizar essa função em outro ponto do programa, "chamando-a" pela mesma letra). O **x** entre parênteses é o *parâmetro* da função, ou seja, o valor a ser for-

- FUNÇÕES EM BASIC
- COMO DEFINIR SUAS PRÓPRIAS FUNÇÕES
- QUANTOS PARÂMETROS?
- COMO USAR UMA FUNÇÃO
- FUNÇÕES PARA MANIPULAÇÃO ALFANUMÉRICA
- UMA FUNÇÃO PARA DESENHAR ELIPSES
- FUNÇÕES MAIS RENDOSAS

necido para que a função efetue os cálculos. Alguns computadores (Apple, TK-2000, MSX e TRS-80) permitem dois ou mais parâmetros, enquanto outros (TRS-Color), apenas um.

Para entender o princípio geral de definição e utilização de uma função em um programa, digite e execute o exemplo a seguir. O programa calcula o cubo de um número qualquer:

```
10 CLS
20 PRINT "NUMERO","CUBO"
30 FOR a=1 TO 20
40 PRINT a,FN c(a)
50 NEXT a
60 DEF FN c(x)=x*x*x
```

```
10 DEF FNC(X)=X*X*X
30 PRINT "NUMERO","CUBO"
40 FOR A=1 TO 13
50 PRINT A,FNC(A)
60 NEXT
```

Uma observação interessante é que, no caso do Spectrum, o computador nunca chega a atingir a linha 60, onde a função **FNC** é definida. Dessa forma, a declaração **DEF FN** funciona como uma linha **DATA**. Nos demais computadores, a linha contendo o **DEF FN** deve ser encontrada pelo programa antes que a função correspondente seja chamada. Assim, é conveniente colocá-la logo no início do programa.

Entretanto, os modelos da linha Spectrum procuram automaticamente por linhas contendo **DEF FN**, toda vez que uma função é invocada. Por isso, costuma-se agrupar todos os **DEF FN** no começo do programa, de modo a dar a este maior velocidade de execução.

COMO CHAMAR UMA FUNÇÃO

Para executar uma função, você tem apenas que digitar seu nome (ou seja, a letra usada em sua definição) logo após a expressão **FN**, e no ponto do programa escolhido para utilização do resultado do cálculo elaborado por essa função. Isso pode ser visto na linha 40 do programa para o Spectrum, e na linha 50 do programa para os demais computadores. No exemplo dado, o programa utiliza **FNC(A)**, em vez de **FNC(X)**, como aparece na definição (linha 60 para o Spectrum, e linha 10 para os demais). No caso, **A** (ou **a** para o Spectrum) é o valor do número cujo cubo queremos obter, ou seja, **A** é o *parâmetro* real da função. Como a variável **x**, usada na **DEF FN** serve apenas para especificar como o parâmetro será usado na expressão de cálculo, ela é chamada *pseudo-parâmetro*.

Portanto, mesmo que o parâmetro seja diferente a cada chamada da função **FNC** em um programa, a fórmula dada permitirá que se faça o cálculo correto e que o resultado seja retornado ao programa exatamente como as outras funções pré-programadas fazem (**SQR**, **COS**, **LOG** etc.). Assim, a função estabelecida pelo usuário pode ser tratada como se fosse uma espécie de *abreviação* para a fórmula de cálculo especificada em sua definição.

No caso de uma função tão simples como a do exemplo, você poderia pensar que seria mais rápido usar a expres-

são X x X x X do que definir a função só para poder usá-la depois. De fato, este é apenas um exemplo. O uso de funções definidas pelo usuário só será rentável quando elas forem chamadas em vários pontos do programa, de preferência com parâmetros que variem. Aí, a economia é realmente notável. Além disso, **DEF FN** é capaz de trabalhar com expressões matemáticas muito mais longas e complexas do que esta.

Uma boa maneira de aprender como funciona **DEF FN** em um programa é experimentar com o exemplo citado, mudando a definição da função de diversas formas (dividindo o parâmetro por 10, por 2000, calculando a raiz quadrada etc.). Para verificar se a função definida por você calculou corretamente, compare os resultados obtidos por ela com os obtidos pelo uso da mesma fórmula, em modo direto.

O NOME DAS FUNÇÕES

Diversas funções podem ser definidas dentro de um mesmo programa. Cada uma delas deve ter um nome, que varia com o computador. Esse nome é geralmente um caractere (ou cadeia de caracteres) colocado imediatamente após o **DEF FN**. Os micros da linha Spectrum só admitem uma letra. Os demais micros podem usar até dois caracteres. Desse modo, o nome de uma função enfrenta as mesmas restrições que um nome de variável: o primeiro caractere deve ser obrigatoriamente uma letra, e o segundo, uma letra ou um número; além disso, não podem ser utilizados caracteres especiais. Se forem usadas mais de duas letras no nome de uma função, o Spectrum reconhecerá apenas as duas primeiras (no exemplo dado, **DEF FNCUBO** será aceita equivalendo a **DEF FNCU**, e não a **DEF FNC**).

DEFINA OS PARÂMETROS

O próximo passo na definição de uma função consiste em adicionar os parâmetros. Estes são formados pelos valores entre parênteses que devem ser fornecidos para que a função efetue o cálculo. Embora possam ser parecidos com variáveis dentro do **DEF FN**, eles são, de fato, apenas pseudovariáveis, que valem só para a definição.

Tudo o que os parâmetros fazem é "dizer" ao computador qual o número de valores a serem usados no cálculo. Em outras palavras, se você começar uma definição deste modo:

```
DEF FNE (A,B,C) =
```

então o computador esperará que o programa forneça três números toda vez que **FNE** for chamada (um erro de sintaxe ocorrerá na maioria dos micros, se um número errado de parâmetros for fornecido ao se chamar a função). Quando a função for chamada, os três parâmetros do exemplo podem ser oferecidos como constantes ou como variáveis.

Embora os parâmetros usados na **DEF FN** não sejam variáveis, você pode encará-los como tal, ou seja, como uma espécie de variável local, que vale apenas na definição. Eles devem ocorrer na expressão após o sinal de igualdade, com o nome com que foram indicados entre parênteses. Exemplo:

```
DEF FNE (A,B,C)=10 + A * B / C
```

Os parâmetros podem ser variáveis numéricas ou alfanuméricas. Mais adiante, veremos algumas aplicações para funções de tratamento de cadeias alfanuméricas.

Já que não são a mesma coisa, parâmetros e variáveis podem ter os mesmos nomes, sem que isso implique em interferências dos primeiros sobre as segundas e vice-versa.

QUANTOS PARÂMETROS?

A declaração **DEF FN** nos micros das linhas Apple, TK-2000 e TRS-Color é menos versátil do que nos demais computadores, pois permite apenas um parâmetro. Todos os outros valores presentes na expressão aritmética restante precisam ser constantes.

Suponhamos que você queira multiplicar um valor qualquer por 9.81, como parte da função que está definindo. O valor a ser multiplicado poderá receber o nome do único parâmetro disponível, por exemplo, A; a expressão de definição da função será 9.81 x A.

Os outros tipos de computador permitem o uso de quantos parâmetros forem necessários, desde que a definição completa da função esteja contida em uma única linha. Quando a função for chamada, deve-se especificar um número igual de parâmetros, como foi estabelecido na definição, mesmo que haja alguns parâmetros a mais que não foram usados na expressão de definição.

Agora, se você quiser verificar esse fato na prática, digite **NEW** e experimente colocar o programa a seguir (não há uma versão para os micros das linhas TRS-Color, Apple e TK-2000):

```
10 DEF FN a(a,b,c,d,e,f)=a*a*
b
20 PRINT "Introduza dois nume
ros"
30 INPUT a,b
40 PRINT FN a(a,b)
```

Quando você tentar executar o programa acima, verificará que ele acusa um erro na linha 40. Experimente então substituí-la por:

```
40 PRINT FNa (a,b,0,0,0,0)
```

O programa passará então a funcionar, embora ele não use nenhuma informação adicional.

POR QUE USAR UMA FUNÇÃO?

A vantagem das funções em relação às expressões torna-se evidente quando é preciso utilizar uma mesma expressão em vários pontos de um programa. Neste caso, o emprego de funções predefinidas economiza memória e tempo de digitação e de execução.

A função definida pelo usuário também é útil quando a expressão é muito grande e passível de erros de digitação.

A seguir, apresentamos um exemplo de função matemática que se pode usar repetidas vezes em um programa. Ela arredonda um número qualquer, fornecido por meio de um **INPUT**, fixando o número desejado de casas decimais. Como essa função requer dois parâmetros, deixamos de oferecer uma versão para o Apple, o TK-2000 e o TRS-Color:

```
10 FOR q=0 TO 1 STEP 0
20 PRINT '''"Introduza o nume
ro a arredondar e quantas cas
as decimais voce   deseja"'''
(Pressione <ENTER> a cada vez
)"
30 INPUT numero,casas
40 PRINT '''numero;" com ";ca
sas;" casas e' ",FN R(numero,
casas)
50 PAUSE 100: NEXT q
60 DEF FN R(a,b)=INT (a*10^b+
0.5)/10^b
```

```
5 DEF FNR(A,B) =  INT (A * 10^B
+ 0.5)/10^B
```

```
10 FOR G=0 TO 1 STEP 0
20 PRINT:PRINT:PRINT
25 PRINT "INTRODUZA O NUMERO A
ARREDONDAR E QUANTAS CASAS
DECIMAIS VOCE DESEJA"
30 INPUT N,CD
40 PRINT:PRINT:PRINT N;" COM ";
CD;" CASAS = ";FNR(N,CD)
50 NEXT G
```

A função **FNR** utiliza uma segunda função em sua definição: é a função **INT**, que obtém a parte inteira de qualquer número. O exemplo mostra também outro aspecto interessante nas funções predefinidas: elas podem abrigar em seu interior outras funções do BASIC, bem como funções definidas pelo usuário. A única limitação, evidentemente, é que não se pode chamar a própria função que está sendo definida.

Embora o TRS-Color, o Apple II e o TK-2000 aceitem apenas funções definidas com um parâmetro, não é difícil fazer bom uso desse recurso mais limitado. Digite e execute o programa a seguir, que define e aplica uma equação para traçar uma elipse. (Ao ser executado, o programa solicitará alguns valores iniciais, como as constantes A e B, e os dois raios da elipse.)

```
10 DEF FNY(X)=SQR(B*R*R-B*X*X/A
)
20 PMODE 3,1
30 PCLS
40 CLS:INPUT"INTRODUZA AS CONST
ANTES A E B ";A,B
50 INPUT "INTRODUZA OS DOIS RAI
OS ";R1,R2
60 R=R1:Y1=FNY(0):R=R2:Y2=FNY(0
)
70 IF Y2>Y1 THEN Y1=Y2
80 IF Y1>95 THEN R1=R1*95/Y1:R2
=R2*95/Y1
90 SCREEN 1,0
100 FOR K=-127 TO 128
110 IF B*R1*R1<B*K*K/A THEN 130
120 R=R1:PSET(K+127,95+FNY(K),2
):PSET(K+127,96-FNY(K),2)
130 IF B*R2*R2<B*K*K/A THEN 150
140 R=R2:PSET(K+127,95+FNY(K),3
):PSET(K+127,96-FNY(K),3)
150 NEXT
160 GOTO 160
```

```
10 DEF FNY(X)=SQR(B*R*R-B*X*X/
   A)
20 HGR
40 HOME:INPUT "INTRODUZA AS CO
   NSTANTES A E B:";A,B
50 INPUT "INTRODUZA OS DOIS RA
   IOS:";R1,R2
60 LET R=R1 : Y1=FNY(0) : R=R2
   : Y2=FNY(0)
70 IF Y2>Y1 THEN Y1=Y2
80 IF Y1>95 THEN R1=R1*95/Y1 :
   R2=R2*95/Y2
100 FOR K=-127 TO 128
110 IF (B*R1*R1)<(B*K*K/A) THEN
   130
115 HCOLOR 3
120 R=R1 : HPLOT K+127,95 + FNY
   (K) : HPLOT K+127,96 - FNY
   (K)
130 IF (B*R2*R2)<(B*K*K/A) THEN
   150
135 HCOLOR 5
140 R=R2 : HPLOT K+127,95 + FNY
   (K) : HPLOT K+127,96 - FNY
   (K)
150 NEXT
```

O programa usa comandos gráficos para desenhar a elipse. A função definida na linha 10 é usada para calcular as coordenadas de cada ponto da elipse. Ela usa internamente uma função matemática predefinida do BASIC: a função de raiz quadrada (**SQR**).

Caso você queira converter esse programa para outros micros, eis aqui a equação das coordenadas da elipse:

$$\frac{X \cdot X}{A} + \frac{Y \cdot Y}{B} = R \cdot R$$

X e Y são as coordenadas horizontal e ver-

tical de cada ponto da elipse, R é o seu raio (na realidade o programa usa dois valores de raio, R1 e R2, para cada elipse), e A e B são constantes que definem o formato da figura.

Alguns microcomputadores, como por exemplo o Spectrum, o MSX e o TRS-80, permitem ainda a utilização de variáveis alfanuméricas em funções definidas pelo usuário.

Se, mesmo depois de tudo o que foi dito, você ainda tem dúvidas a respeito da utilidade de uma função que manipula cadeias alfanuméricas, digite e execute o seguinte programa:

```
10 PRINT "Introduza seu nome"
20 INPUT n$
30 IF CODE n$>90 THEN  LET n$
=FN u$(n$)
40 PRINT ''"Ola,";n$;" - eu n
ao sou esperto ?"
90 DEF FN u$(x$)=CHR$ (CODE x
$-32)+x$(2 TO )
```

```
5 DEF FNU$(X$) = CHR$ (ASC(X$)
 - 32)+MID$(X$,2)
10 INPUT "INTRODUZA  SEU PRIMEI
RO NOME",N$
30 IF ASC(N$)>90 THEN N$ = FNU$
(N$)
40 PRINT:PRINT "OLA, ";N$;", EU
NAO SOU ESPERTO ?"
```

Observe que esse programa permite inicialmente que você entre um nome; em seguida, ele chama uma função definida pelo usuário; essa função imprimirá sucessivamente a primeira letra em maiúscula e as restantes em caracteres minúsculos.

Funções alfanuméricas como essa trabalham da mesma maneira que funções puramente matemáticas. Seus nomes, porém, são diferentes.

Diversamente das funções matemáticas, esses nomes devem conter sempre uma letra seguida de um cifrão, exatamente como acontece com os nomes de variáveis alfanuméricas.

Como se pode perceber pelo exemplo, a função pode ter parâmetros alfanuméricos, que são usados da mesma forma que parâmetros numéricos.

Um outro programa exemplifica melhor este ponto. Trata-se de uma rotina que emprega uma função alfanumérica para abrilhantar as cadeias entradas por meio de comandos **INPUT**:

```
10 FOR f=0 TO 1 STEP 0
20 INPUT "Introduza o titulo
";i$
30 CLS
35 PRINT FN t$(" "+i$)
40 PAUSE 150
50 NEXT f
60 DEF FN t$(x$)=CHR$ 18+CHR$
1+CHR$ 16+CHR$ 2+"***********
**********************"+CHR$
23+CHR$ (16-LEN x$/2)+x$+CHR$
 23+CHR$ 31+"****************
******************"
```

```
5 DEF FNT$(X$) = STRING$(16-LEN
(X$)/2)+X$+STRING$(16-LEN (X$)
/2)
10 FOR F=0 TO 1 STEP 0
20 INPUT "INTRODUZA O TITULO ";
I$
30 CLS
40 FOR I=1 TO 1000:NEXT I
50 NEXT F
```

Esse programa apresenta algumas particularidades que devem ser especialmente consideradas.

Note, por exemplo, o valor um pouco estranho do comando **STEP** na linha 10. Na verdade, o que ele faz é causar um laço de repetição infinita, informando diretamente ao computador que o incremento do comando **FOR** é 0; isso quer dizer que ele nunca irá alcançar o valor final 1.

Embora neste caso seja mais fácil recorrer a um comando **GOTO**, é conveniente que você saiba que programas melhor estruturados evitam o **GOTO** para trás, o que seria considerado má prática de programação.

FUNÇÕES RENDOSAS

Eis aqui outro exemplo do comando **DEF FN** em ação. Trata-se de um programa relativamente simples, que pode ser empregado em tarefas como calcular os juros de sua caderneta de poupança.

Ele define uma função que calcula rigorosamente o valor a ser atingido por um investimento após um dado período de tempo, e levando em consideração os juros pagos por ano.

```
10 DEF FN c(t)=INT (AM*((R/
100+1)^T)*100)/100
20 INPUT "Quantos cruzados? "
;AM
30 INPUT "Taxa de juros (%)?
";R
40 INPUT "No. de unidades de
tempo? ";T
50 PRINT AT 9,0;"Quantia tota
```

```
1 apos a operacao= ";TAB 12;
FN C(T)
60 PRINT  INVERSE 1;AT 20,1;"
Qualquer tecla para recomecar
"
70 PAUSE 0: CLS : GOTO 20
```

```
10 DEF FN C(T)=INT(AM*((1+R/100
)^T)*100)/100
20 INPUT"QUANTOS CRUZADOS ";AM
30 INPUT"TAXA DE JUROS (%) ";R
40 INPUT"QUANTAS UNIDADES DE TE
MPO ";T
50 PRINT"QUANTIA TOTAL APOS APL
ICACAO =",FNC(T)
60 PRINT:PRINT"QUALQUER TECLA P
ARA RECOMECAR"
70 A$=INKEY$:IF A$="" THEN 70
80 GOTO 20
```

Para o programa funcionar no Apple e no TK-2000, substitua a linha 70 por:

```
70 GET A$
```

O programa funciona de forma muito simples. Definida a função, os valores de entrada são colocados por meio de uma série de comandos **INPUT**, de modo que o usuário possa fornecer ao computador os detalhes da poupança.

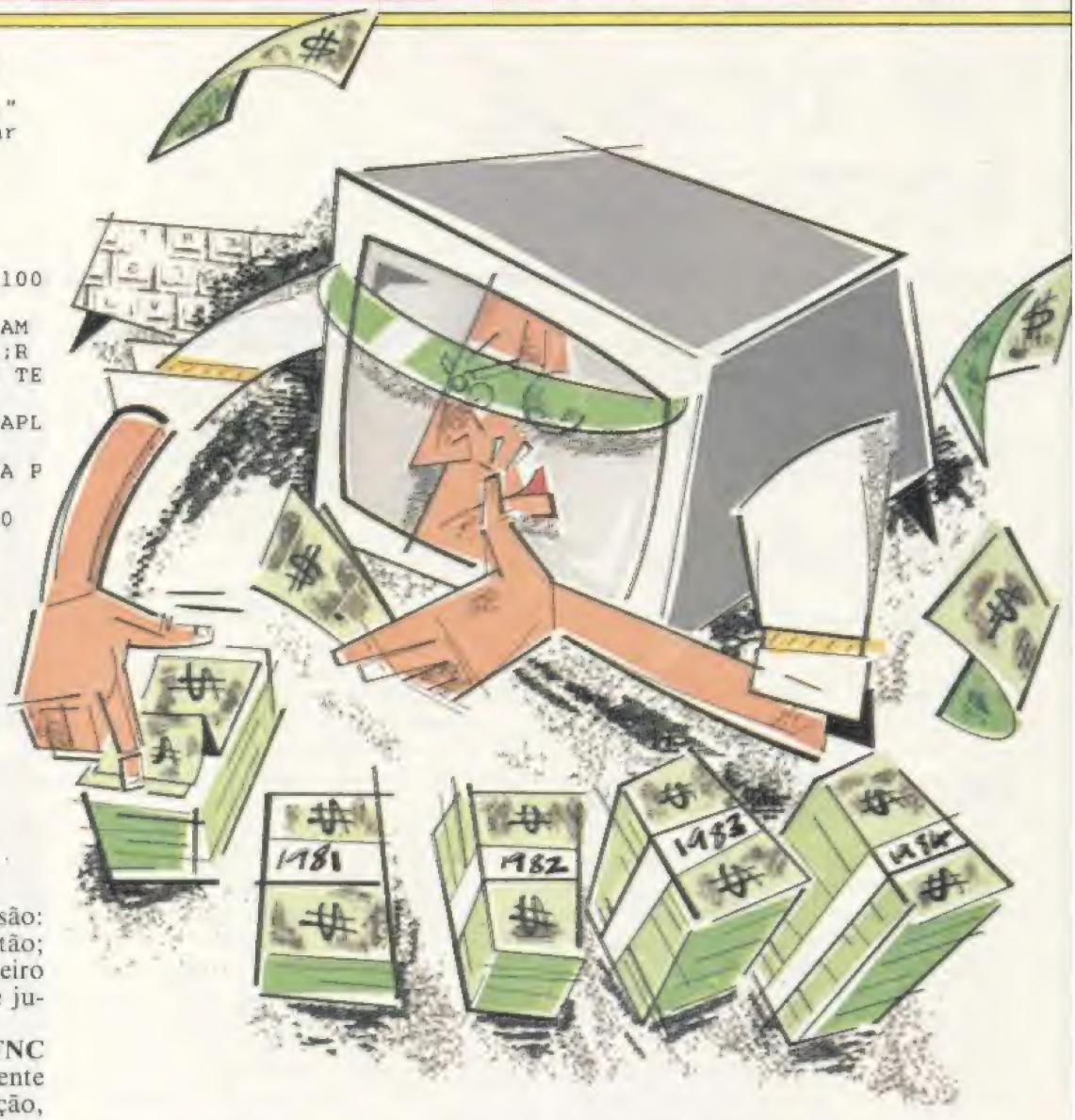
As variáveis que o programa usa são: **T**, para o período de tempo em questão; **AM**, para a quantia inicial em dinheiro a ser aplicada; e **R**, para a taxa de juros, em % ao ano.

O único parâmetro da função **FNC** é **T**. Isso pode ser percebido facilmente a partir da própria definição da função, na linha 10. Entretanto, a expressão matemática na mesma linha contém três variáveis, que são **AM**, **R** e **T** — e apenas uma delas é o parâmetro. Esta é uma característica muito útil das funções definidas pelo usuário. Ela permite que se misturem parâmetros e variáveis na mesma expressão, evitando que seja necessário definir todos os elementos como parâmetros.

Esta é uma maneira de se contornar a limitação de um parâmetro por função nos microcomutadores das linhas TRS-Color, Apple e TK-2000.

Já no MSX é possível até mesmo definir uma função inteiramente sem parâmetros (neste caso, não coloque parênteses, nem em **DEF FN**, nem na chamada à função).

Os usuários do Spectrum podem estranhar o fato de que a expressão na linha 10 usa variáveis ainda não definidas, o que deveria provocar uma mensagem de erro. Entretanto, como foi dito linhas atrás, no Spectrum o processo funciona de modo diferente; neste caso, a linha 10 é procurada *depois* que as variáveis foram definidas pelos comandos **INPUT** do programa.



Os juros calculados aqui são juros compostos — isso quer dizer que o valor ganho em um período é usado para calcular os juros do período seguinte (juros sobre juros).

OPERAÇÕES ARITMÉTICAS

Isso é levado em consideração automaticamente pela fórmula da função, simplificando o cálculo dos números procurados. Veja a seguir como se processa esse cálculo na prática.

Se, por exemplo, a taxa de juros for 10% após um período de tempo, o programa calculará o valor original mais 0.1 multiplicado por esse valor. Você pode ver isto na expressão contida na função, como R/100+1. Assim, no exemplo dos 10%, o resultado após um período de aplicação é de 1.1 vezes o valor original aplicado. Após dois períodos, será 1.1 vezes o novo valor, e assim por diante. Em **T** períodos, isso dará 1.1 elevado a **T**. Esta é a segunda parte da função, que toma a forma completa vista no programa, ou seja, **(R/100+1)T**. A parte restante da expressão multiplica isto pelo valor do principal (**AM**) e arredonda o resultado final para duas casas decimais, multiplicando por 100, tomando o seu **INT**eiro, e dividindo por 100.

Tudo se resume, portanto, a simples operações aritméticas.

Você poderá também definir funções trigonométricas que não existem na maioria dos interpretadores BASIC, como arco seno e arco co-seno.

```
ASN(X)=ATN(X/SQR(-X*X+1))
ACS(X)=-ATN(X/SQR(-X*
X+1))+1.5708
```

Agora, procure treinar, "inventando" algumas funções novas para outras áreas da matemática.

UM EDITOR DE TEXTOS (3)

A procura da palavra certa foi sempre um elemento estimulante para os grandes escritores. Esta lição ensina como fazer isso com a prosaica ajuda de um computador.

As duas primeiras partes desta lição apresentaram as funções básicas de edição na tela, que permitem criar textos e armazená-los. A terceira e última parte fornece as rotinas de ordenação, procura de palavras, impressão e uso de "máscara de texto".

ORDENAÇÃO

Baseada na rotina de troca retardada, a rotina de ordenação é usada para colocar linhas em ordem alfabética. Ela é de grande utilidade para ordenar nomes ou outros dados.

PROCURA DE PALAVRAS

Essa rotina "varre" o texto em busca de uma determinada palavra ou fragmento de palavra. Ela deve ser ativada dentro do modo editor. A procura começa no ponto onde o marcador está localizado. Por isso, certifique-se de que ele está na posição mais adequada para dar início à busca.

Há dois motivos básicos para que não se ache a palavra procurada: ou ela foi digitada de modo errado ou está dividida em duas linhas.

Quando a palavra for encontrada, o programa continuará no modo editor e a linha poderá ser copiada imediatamente na área de trabalho.

IMPRESSÃO

A rotina de impressão possibilita o envio do texto gerado para a impressora. Ela apresenta algumas características especiais, incluindo uma rotina para controlar o formato das cartas e outra para uso do texto com "máscara", permitindo que você adicione partes variáveis ao texto.

Se a sua impressora trabalha com comandos não padronizados, pode ser necessário algum ajuste no programa. Preste particular atenção aos caracteres que provocam avanço de linha (*linefeed*, ou **LF**) e retorno do carro (*carriage return*, ou **CR**). O programa usa o caractere ASCII 13 para ambas as funções.

FORMATAÇÃO

Pouco adiantaria poder escrever e corrigir um texto se não fosse possível imprimi-lo na forma desejada. Pode ser necessário, por exemplo, colocar o cabeçalho da página no centro de uma linha, seguida por outras linhas em branco. Isso se torna fácil quando se usam os comandos de formatação. Outro exemplo comum é quando se quer colocar a data ou o endereço do remetente na margem direita e o endereço do destinatário à esquerda.

Os símbolos empregados na rotina de formatação são os mesmos do programa do artigo *Escreva Cartas sem Esforço* (página 17). Lembre-se de que eles devem aparecer sempre no início da linha. Esses símbolos são os seguintes: o sustenido ("#"), que coloca a linha de texto na margem direita da página (se existir só uma linha com essa marca, ela será colocada com o último caractere na última posição da linha; se houver mais linhas, o programa procurará a mais longa e alinhará as outras com base nesta); o ampersand ("&"), que faz o texto que o segue ser impresso no início da linha seguinte (por exemplo, ele pode ser usado em cartas, para posicionar cada linha do endereço do destinatário); o cifrão ("$"), que, além de passar o texto para outra linha, provoca um recuo na primeira palavra da nova linha; o asterisco ("*"), que coloca o texto no centro da linha. Ao usar essa rotina, tenha cuidado com linhas de texto muito longas. Estas devem ter, no máximo, a mesma largura da linha de impressão.

COMO USAR "MÁSCARA DE TEXTO"

Além dos comandos de formatação, o programa dispõe de outro recurso (inexistente no do Spectrum) com o qual se pode criar uma carta padrão e deixar que algumas partes específicas sejam variáveis. Esse recurso é o "símbolo de máscara". Ele opera do seguinte modo: quando você quiser inserir uma frase no texto, coloque um par de colchetes ("][") na posição escolhida. No momen-

- ORDENAÇÃO
- A ARTE DE PROCURAR PALAVRAS
- IMPRESSÃO
- CARTA PADRÃO
- FORMATAÇÃO
- CENTRAR O TEXTO
- AJUSTAR À DIREITA
- AJUSTAR À ESQUERDA
- ESPAÇAMENTO

to da impressão, esse sinal será substituído pelo bloco de texto apropriado. Dessa maneira, você pode começar a carta com um "Caro][" e só digitar o nome do destinatário quando lhe convier.

O "símbolo de máscara" pode ser colocado em qualquer posição no texto, menos depois de um sustenido (#). É que o programa mede primeiro o tamanho de cada linha que vai ser posicionada à direita, determinando a seguir onde deve começar a impressão dessas linhas. Como essa operação é feita antes de se carregar os blocos variáveis, sempre surge algum problema.

O texto que substitui os colchetes é digitado diretamente do teclado na hora da impressão, ou lido de um arquivo criado pelo próprio programa.

O máximo de caracteres admitido por linha no TRS-Color é 32. No MSX, esse número sobe para 39 e, no Apple, para 40. O texto será quebrado, assim, em unidades de 32 a 40 caracteres, dependendo da máquina. Cada par de colchetes permite a entrada de, no máximo, 250 caracteres.

Se os blocos variáveis de texto forem carregados de um arquivo, o computador se encarregará de colocar os dados nos seus devidos lugares. Cuide apenas para que os dados estejam na ordem correta e para que haja informação para todos os pares de colchetes.

```
4000 REM imprimir
4010 LET tt=(pl-ll)/2
4020 LET d=0
4025 FOR n=t+3 TO b-3
4030 LET a$=t$(n)
4032 IF LEN a$=0 THEN  NEXT n:
RETURN
4034 IF a$(LEN a$-1)<>CHR$ 32 T
HEN  GOTO 4037
4035 IF a$(LEN a$)=CHR$ 32 THEN
  LET a$=a$( TO LEN a$-1): GOTO
 4032
4037 LET l=LEN a$
4040 LET c=0
4050 IF c=l THEN  NEXT n: LPRIN
T CHR$ 13: RETURN
4060 LET c=c+1: LET d=d+1: IF c
>l THEN  GOTO 4100
4070 IF a$(c)="#" THEN  GOTO 45
00
4080 IF a$(c)="*" THEN  GOTO 47
00
4085 IF a$(c)="&" THEN  GOTO 48
50
4090 IF a$(c)="$" THEN  LPRINT
CHR$ 13;CHR$ 13;: LET d=0: GOTO
 4900
4100 LET n=n+1: IF n>=b-1 THEN
 LET l=LEN a$: GOTO 4111
4105 IF t$(n,1)="$" OR t$(n,1)=
"#" OR t$(n,1)="*" OR t$(n,1)="
&" THEN  GOTO 4110
4106 LET a$=a$+t$(n)
4107 IF a$(LEN a$-1)<>CHR$ 32 T
HEN  GOTO 4100
4108 IF a$(LEN a$)=CHR$ 32 THEN
  LET a$=a$( TO LEN a$-1)
4109 GOTO 4107
4110 LET n=n-1: LET l=LEN a$
4111 IF a$(c)=CHR$ 32 THEN  GOT
O 4800
4112 LPRINT a$(c);
4115 IF d>ll THEN  LET d=0
4120 GOTO 4050
4500 LET nl=0: LET ta=ll: LET b
e=0
4510 LET le=LEN a$-1: IF le>ll
THEN  PRINT  FLASH 1;"ERRO NO F
ORMATO - ENDERECO MUITOLONGO":
SOUND 2,10: RETURN
4520 IF le>be THEN  LET be=le
4530 LET nl=nl+1: LET n=n+1: LE
T a$=t$(n)
4532 IF LEN a$=0 THEN  NEXT n:
RETURN
4535 IF a$(LEN a$)=CHR$ 32 THEN
  LET a$=a$( TO LEN a$-1): GOTO
 4532
4538 IF a$(1)="#" THEN  GOTO 45
10
4540 LET n=3
4550 LET tr=tt+ll-be: FOR q=1 T
O nl: FOR h=1 TO tr: LPRINT CHR
$ 32;: NEXT h: LET n=n+1: LET a
$=t$(n)
4552 IF LEN a$=0 THEN  NEXT n:
RETURN
4555 IF a$(LEN a$)=CHR$ 32 THEN
  LET a$=a$( TO LEN a$-1): GOTO
 4552
4558 LPRINT a$(2 TO ): NEXT q
4560 NEXT n: RETURN
4700 LET ta=(ll-1)/2+tt: IF ta<
tt THEN  LPRINT CHR$ 13: PRINT
 FLASH 1;"ERRO NO FORMATO - IMP
OSSIVEL    CENTRALIZAR": SOUND
2,10: RETURN
4710 LPRINT CHR$ 13;: FOR m=1 T
O ta: LPRINT CHR$ 32;: NEXT m:
LPRINT a$(2 TO );: LET d=0: NEX
T n: RETURN
4800 LET sl=ll-d-1: LET cc=c+1:
```

```
 LET x=1
4810 IF cc>=1 THEN  GOTO 4825
4820 IF a$(cc)<>CHR$ 32 THEN  L
ET cc=cc+1: LET x=x+1: GOTO 481
0
4825 IF x>=11 THEN  LPRINT CHR$
 13: PRINT  FLASH 1;"ERRO NO FO
RMATO - PALAVRA MUITO GRANDE":
SOUND 2,10: RETURN
4830 IF s1>=x THEN  GOTO 4112
4850 LPRINT CHR$ 13;: LET d=0
4900 FOR m=1 TO tt: LPRINT CHR$
 32;: NEXT m: GOTO 4050
8000 REM procura
8002 IF z$="" THEN  PRINT #1;AT
 0,0; BRIGHT 1;"Nao foi definid
a a palavra alvo": PAUSE 100: P
RINT #1;AT 0,0;s$;s$: RETURN
8005 PRINT #1;AT 0,0;s$;s$: IF
p=b-2 THEN  LET p=4
8010 FOR n=1 TO 33-LEN z$
8020 IF t$(p,n TO n+LEN z$-1)=z
$ THEN  LET n=33-LEN z$: NEXT n
: GOTO 8050
8030 NEXT n
8040 LET p=p+1: IF p=b-2 THEN
LET p=p-1: GOTO 8050
8045 GOTO 8010
8050 LET p=p+1: GOSUB 1000: RET
URN
8500 REM ordenacao
8505 PRINT #1;AT 0,0;s$;s$
8510 LET ss=4
8520 IF t$(ss,1)="^" THEN  GOTO
 8550
8530 LET ss=ss+1: IF ss=b THEN
 PRINT #1;AT 0,0; BRIGHT 1;"Lim
ites nao definidos": PAUSE 100:
 PRINT #1;AT 0,0;s$;s$: RETURN
8540 GOTO 8520
8550 LET se=ss+1
8560 IF t$(se,1)="^" THEN  GOTO
 8600
8570 LET se=se+1: IF se=b THEN
 PRINT #1;AT 0,0; BRIGHT 1;"Som
ente um limite foi definido": P
AUSE 100: PRINT #1;AT 0,0;s$;s$
: RETURN
8580 GOTO 8560
8600 IF ss=se-1 OR ss=se-2 THEN
  GOTO 8900
8610 PRINT #1;AT 0,0; BRIGHT 1;
"Ordenando"
8620 FOR i=ss+1 TO se-1
8630 LET k=i
8640 FOR j=i+1 TO se-1
8650 IF t$(j)<t$(k) THEN  LET k
=j
8660 NEXT j: IF u<>k THEN  LET
w$=t$(k): LET t$(k)=t$(i): LET
t$(i)=w$
8670 NEXT i
8900 FOR n=ss TO b: LET t$(n)=t
$(n+1): NEXT n
8910 FOR n=se-1 TO b: LET t$(n)
=t$(n+1): NEXT n: LET b=b-2: IF
 p>b-2 THEN  LET p=p-2
8915 PRINT #1;AT 0,0;s$;s$
8930 GOSUB 1000
8940 RETURN
```

Para usar a rotina de ordenação, marque os pontos correspondentes ao início e ao fim da operação, colocando uma flecha (↑) acima da primeira linha a ser ordenada e outra abaixo da última. Para iniciar o processo, tecle **[CAPS SHIFT]** e 4. As flechas serão eliminadas durante a ordenação.

Para fazer a operação de busca, tecle **[CAPS SHIFT]** e 2; a palavra a ser procurada deve ser digitada conforme requisitado. A busca começará imediatamente. Assim que o computador encontrar a primeira ocorrência, o cursor aparecerá sob ela. Se quiser continuar a procura mais adiante, pressione **[CAPS SHIFT]** e 3. Para alterar a palavra, tecle **[CAPS SHIFT]** e 2.

Tecle 6 para entrar na rotina de impressão da linha 4000. Isso fará o texto ser impresso de acordo com o que foi definido na rotina da linha 100, ou seja, 32 caracteres por linha e 32 linhas por página. Se quiser alterar esses valores, tecle 7 e não 6.

Na formatação do texto, coloque o sinal adequado, para que a linha seja impressa do modo desejado. O sinal "#" indica que a linha deve ser posicionada junto à margem direita do texto. O "&" provoca uma mudança de linha e inicia a impressão na margem esquerda. O "$" faz com que haja um avanço de duas linhas e não de uma apenas. O "*" centraliza o texto.

```
3000 CLS:PRINT@7,BL$;"IMPRESSAO
";BL$
3010 IF TL<2 THEN 3050
3020 PRINT"DA (M)EMORIA OU DE (
A)RQUIVO ?"
3030 R$=INKEY$:IF R$<>"M" AND R
$<>"A" THEN 3030
3040 IF R$="M" THEN 3060
3050 GOSUB 4500
3060 IF TL=1 THEN PRINT (nao ha
 arquivo":PY$="T2003EDCA":GOTO
3570
3070 KF=0:PRINT "QUER USAR MASC
ARA (S/N) ?"
3080 R$=INKEY$:IF R$<>"S" AND R
$<>"N" THEN 3080
3090 IF R$="N" THEN 3150
3100 PRINT:PRINT"CARREGA BLOCOS
 VARIAVEIS          (T)ECLADO OU
```

```
(A)RQUIVO?"
3110 R$=INKEY$:IF R$<>"T" AND R
$<>"A" THEN 3110
3120 KF=2:IF R$="T" THEN KF=1:G
OTO 3150
3130 PRINT:LINEINPUT "INTRODUZA
 O NOME DO ARQUIVO ";VB$
3140 IF LEFT$(VB$,1)<"A" OR LEF
T$(VB$,1)>"Z" THEN 3130
3150 CLS:PRINT "DESEJA REPROGRA
MAR A IMPRESSORA (S/N) ?"
3160 R$=INKEY$:IF R$<>"S" AND R
$<>"N" THEN 3160
3170 IF R$="S" GOSUB 5500
3180 CLS
3190 VB=0:PP=0:AZ=0:LC=1:PRINT
"DESEJA IMPRIMIR NA TELA (S/N)?
":PRINT"enter RETORNA AO MENU P
RINCIPAL"
3200 R$=INKEY$:IF R$<>"S" AND R
$<>"N" AND R$<>CHR$(13) THEN 32
00
3210 IF R$=CHR$(13) THEN RETURN
3220 IF KF=0 THEN 3240
3230 IF DL=1 AND KF=2 THEN FREA
D VB$,FROM 0;DV: FREAD VB$;DV E
LSE IF K=2 THEN OPEN "I",#-1,VB
$:INPUT#-1,DV,DV
3240 P=0:GP$="":IF R$="N" THEN
P=-2:GP$=STRING$(GP,32)
3250 FOR K=1 TO TL-1:IF LEFT$(T
X$(K),1)="#" AND LEN(TX$(K))-1>
AZ THEN AZ=LEN(TX$(K))
3260 NEXT:IF AZ>TW THEN PRINT"e
rro-endereco muito longo":PY$="
T402AB":GOTO 3570
3270 K=1:PRINT#P,LF$;GP$;:A$=""
:IF AZ>0 THEN AZ$=STRING$(GP+TW
-AZ,32)
3280 TT$=TX$(K)
3290 IF TT$="" THEN PRINT #P,CH
R$(13);GP$;:PP=0:LC=LC+1:GOSUB
3590:GOTO 3520
3300 BP=INSTR(TT$,"][");IF BP=0
 OR KF=0 THEN 3390
3310 IF KF=1 THEN 3370
3320 IF DL=1 THEN 3360
3330 IF EOF(-1) THEN 3350
3340 INPUT#-1,RP$:GOTO 3380
3350 PRINT"erro-falta de dados
no arquivo":PY$="L2005DL402D":G
OTO 3570
3360 IF EOF(VB$) THEN 3350 ELSE
 FLREAD VB$;RP$:GOTO 3380
3370 BL=BL+1:PRINT:PRINT "INTRO
DUZA BLOCO VARIAVEL";BL;"?";:LI
NEINPUT RP$
3380 TT$=LEFT$(TT$,BP-1)+RP$+MI
D$(TT$,BP+2):GOTO 3300
3390 ON INSTR ("&$*#",LEFT$(TT$
,1)) GOTO 3460,3470,3490,3510
3400 IF PP+LEN(TT$)<=TW THEN PR
INT#P,TT$;:PP=PP+LEN(TT$):GOTO
3520
3410 TA$=LEFT$(TT$,TW-PP)
3420 IF INSTR(TT$," ")>TW THEN
PRINT "erro-palavra muito grand
e na",TT$:PY$="T1002CB":GOTO 35
70
3430 IF RIGHT$(TA$,1)=" " THEN
3450
3440 IF LEN(TA$)>0 THEN TA$=LEF
T$(TA$,LEN(TA$)-1):GOTO 3430
3450 PRINT#P,TA$;CHR$(13);GP$;:
PP=0:LC=LC+1:GOSUB 3590:TT$=MID
$(TT$,LEN(TA$)+1):IF TT$<>"" TH
EN BP=1:GOTO 3400 ELSE 3520
3460 PRINT#P,CHR$(13);GP$;:PP=0
:LC=LC+1:GOSUB 3590:TT$=MID$(TT
$,2):GOTO 3300
3470 TT$=MID$(TT$,2):PRINT#P,CH
R$(13);GP$;STRING$(INT(TW/4),32
);:PP=INT(TW/4)
3480 LC=LC+1:GOSUB 3590:GOTO 33
00
3490 TT$=MID$(TT$,2):IF LEN(TT$
)>TW THEN PRINT"erro-impossivel
 centralizar";TT$:PY$="T103C":G
OTO 3520
3500 PRINT#P,CHR$(13);GP$;STRIN
G$(INT((TW-LEN(TT$))/2),32);TT$
;CHR$(13);GP$;:PP=0:LC=LC+1:GOS
UB 3590:GOTO 3520
3510 PRINT#P,CHR$(13);AZ$;MID$(
TT$,2);:PP=0:LC=LC+1:GOSUB 3590
3520 K=K+1:IF P=0 THEN FOR Z=1
TO 500:NEXT
3530 IF K<TL THEN 3280
3540 IF P=-2 THEN PRINT #P,LF$;
LF$ ELSE PRINT:PRINT
3550 IF K=1 THEN CLOSE #-1 ELSE
 IF KF=2 THEN CLOSE
3560 IF P=0 THEN 3190 ELSE RETU
RN
3570 FOR Z=1 TO 10:PLAY PY$:NEX
T:IF KF=1 THEN CLOSE#-1 ELS IF
KF=2 THEN CLOSE
3580 RETURN
3590 IF LC>TH THEN PRINT #P,LF$
;LF$;GP$;:LC=1
3600 RETURN
5070 L=CP:PRINT @384,"INTRODUZA
 PALAVRA PROCURADA"
5080 LINEINPUT TG$:IF TG$="" TH
EN 5070
5090 PRINT @500,"procurando";BL
$;
5100 IF L=TL THEN CP=TL:CLS:GOS
UB 2090:RETURN
5110 IF INSTR(TX$(L),TG$)=0 THE
N L=L+1:GOTO 5100
5120 CP=L+1:CLS:GOSUB 2090:RETU
RN
5130 IF SS>SE THEN TT=SS: SS=SE
: SE=TT
5140 SE=SE-1
5150 PRINT@500,"ordenando";BL$;
5160 FOR I=SS TO SE-1
5170 K=I
5180 FOR J=I+1 TO SE
5190 IF TX$(J)<TX$(K) THEN K=J
5200 NEXT:IF I<>K THEN TT$=TX$(
K):TX$(K)=TX$(I):TX$(I)=TT$
5210 NEXT:CLS:GOSUB 2090:RETURN
```

Para fazer funcionar a rotina de ordenação, entre no modo editor e leve o marcador ">" para um dos extremos do bloco a ser ordenado. Então, tecle "@". Em seguida, mova o marcador, levando-o para o outro extremo, e tecle "@" novamente. Essa operação inicia automaticamente a ordenação.

A função de procura de palavras é ativa, dentro do modo editor, quando se pressiona a tecla P. Você responde qual a palavra a ser procurada por intermédio da mensagem mostrada na área de trabalho. Pressione a tecla [**RETURN**] para dar início à busca. Quando a palavra desejada for encontrada, o marcador será colocado logo abaixo da linha que a contém.

Para imprimir um texto já editado, tecle I a partir do menu principal. Você terá então de responder a várias perguntas. A primeira delas indaga se o texto a ser impresso está na memória ou no arquivo. Caso você escolha imprimir o que está na memória e não haja nada aí, um sinal sonoro será ativado e uma mensagem mostrada na tela do computador; o programa voltará então ao menu principal.

Em seguida, será oferecida a opção de mudar os parâmetros da impressora. Responda com S ou N. Essa rotina torna possível a alteração dos valores predeterminados, que são os seguintes: 80 para a largura do formulário, 60 para a largura da linha de texto (deixando margens de dez colunas), 66 para o número de linhas total por formulário e 60 para o número de linhas de texto (deixando três linhas de espaço no topo e no fim da página).

Assumidos quando o programa é executado, os valores indicados acima serão mantidos até que você os altere.

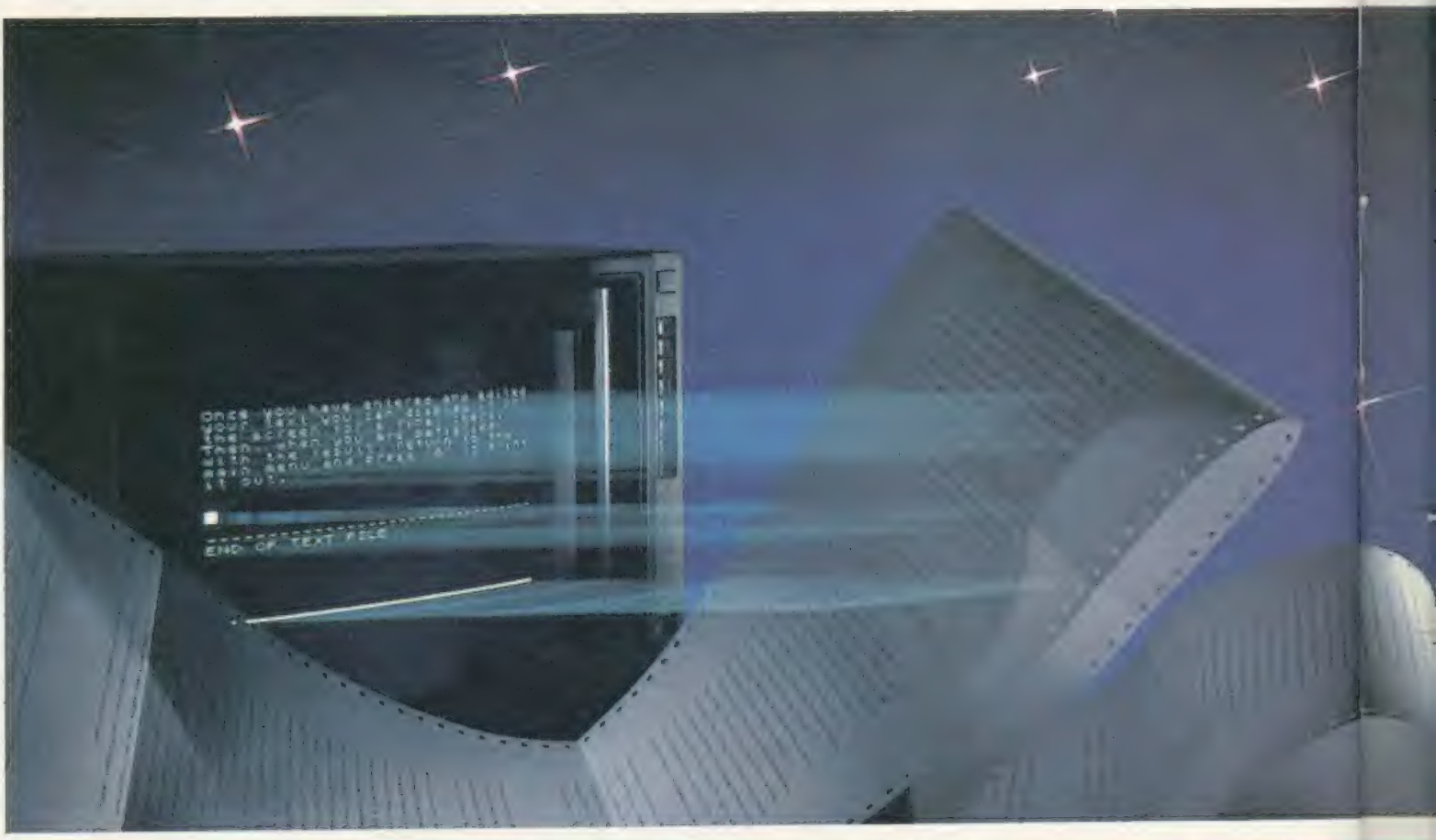
A última pergunta diz respeito a uma cópia do texto na tela. Se você responder S, o texto será imediatamente colocado na tela; um N se encarrega de enviá-lo para a impressora.

Os sinais empregados para formatação do texto são os mesmos que aparecem no programa do já citado artigo da página 17. O sustenido ("#") faz com que a linha seja levada para a margem direita, enquanto o cifrão ("$") provoca um avanço de uma linha e um recuo do texto subseqüente.

O ampersand ("&") determina um avanço de linha, fazendo com que o texto a seguir seja impresso no início da linha abaixo (sem recuo). O asterisco ("*") centraliza a linha de texto.

Para inserir blocos de texto variáveis dentro do texto principal, utilize um par de colchetes ("][") na posição desejada. Responda S quando perguntado se deseja recorrer à "máscara de texto". Os blocos podem ser lidos a partir do teclado ou de um arquivo. Se sua opção for pelo teclado você deve digitar cada bloco quando o programa pedir. No caso de escolher o arquivo, essa operação será feita automaticamente — o computador lerá o texto de um arquivo previamente preparado para isso (use o próprio programa).

```
3000 CLS:BL$="Imprimir":GOSUB 2
220
3010 IF TL<2 THEN 3050
3020 PRINT:PRINT:PRINT"Imprime
texto da [M]emória ou [F]ita?";
3030 R$=INKEY$:IF R$<>"M" AND R
$<>"F" THEN 3030
3040 IF R$="M" THEN 3060
3050 GOSUB 4500:CLS:BL$="Imprim
ir":GOSUB 220
3060 IF TL=1 THEN BEEP:LOCATE 8
,12:PRINT"Não há texto na memór
ia!":FOR I=1 TO 1000:NEXT:RETUR
N
3070 KF=0:PRINT:PRINT"Quer usar
 máscara? (S/N)";
3080 R$=INKEY$:IF R$<>"S" AND R
$<>"N" THEN 3080
3090 IF R$="N" THEN 3150
3100 PRINT:PRINT"Carrega os blo
cos variáveis do          [T]ecla
do ou da [F]ita?";
3110 R$=INKEY$:IF R$<>"T" AND R
$<>"F" THEN 3110
3120 KF=2:IF R$="T" THEN KF=1:G
OTO 3150
3130 PRINT:LINEINPUT"Nome do ar
quivo ";VB$
3140 IF LEFT$(VB$,1)<"A" OR LEF
T$(VB$,1)>"Z" THEN 3130
3150 PRINT:PRINT"Altera a confi
guração                   da imp
ressora? (S/N)";
3160 R$=INKEY$:IF R$<>"S" AND R
$<>"N" THEN 3160
3170 IF R$="S" THEN GOSUB 5500
3180 CLS:BL$="Imprimir":GOSUB 2
220
3190 VB=0:PP=0:AS=0:LC=1:PRINT:
PRINT:PRINT"Quer uma demonstraç
ão na tela? (S/N)":PRINT"Tecle
<RETURN> para retornar ao menu"
;
3200 R$=INKEY$:IF R$<>"S" AND R
$<>"N" AND R$<>CHR$(13) THEN 32
00
3210 IF R$=CHR$(13) THEN RETURN
3220 IF KF<>2 THEN 3240
3230 OPEN VB$ FOR INPUT AS #1:I
NPUT#1,DV,DV
3240 P=0:GP$="":IF R$="N" THEN
P=-1:GP$=STRING$(GP,32)
3250 FOR K=1 TO TL-1:IF LEFT$(T
X$(K),1)="#" AND LEN(TX$(K))-1>
AS THEN AS=LEN(TX$(K))
3260 NEXT:IF AS>TW THEN PRINT:P
RINTTAB(10);"Endereço muito lon
go!":GOTO 3610
3270 K=1:A$="":IFAS>0 THEN A$=S
TRING$(GP+TW-AS,32)
3280 IF NOT P THEN PRINT:PRINT:
PRINT;ELSE LPRINT LF$;GP$;
3290 TT$=TX$(K)
3300 IF TT$<>"" THEN 3320 ELSE
IF P THEN LPRINT L1$;GP$ ELSE P
RINT
3310 PP=0:LC=LC+1:GOSUB 3630:GO
TO 3560
3320 BP=INSTR(TT$,"][");IF BP=0
 OR KF=0 THEN 3390
3330 IF KF=1 THEN 3370
3340 IF EOF(1) THEN 3360
3350 INPUT#1,RP$:GOTO 3380
3360 PRINT:PRINTTAB(3)"Fim de d
ados para blocos variáveis!":GO
TO 3610
3370 BL=BL+1:PRINT:PRINT"Bloco
variável";BL;"?";:LINEINPUT RP$
3380 TT$=LEFT$(TT$,BP-1)+RP$+MI
D$(TT$,BP+2):GOTO 3320
3390 ON INSTR("&$*#",LEFT$(TT$,
1)) GOTO 3470,3490,3510,3540
3400 IF PP+LEN(TT$)<=TW THEN IF
 P THEN LPRINT TT$;CHR$(32);:PP
=PP+LEN(TT$):GOTO 3560 ELSE PRI
NTTT$;CHR$(32);:PP=PP+LEN(TT$):
GOTO 3560
3410 TA$=LEFT$(TT$,TW-PP)
3420 IF INSTR(TT$," ")>TW THEN
PRINT:PRINT"Palavra muito longa
 em:":PRINTTT$:GOTO 3610
3430 IF RIGHT$(TA$,1)=CHR$(32)
THEN 3450
3440 IF LEN(TA$)>0 THEN TA$=LEF
T$(TA$,LEN(TA$)-1):GOTO 3430
3450 IF P THEN LPRINT TA$;L1$;G
P$; ELSE PRINTTA$
3460 PP=0:LC=LC+1:GOSUB 3630:TT
$=MID$(TT$,LEN(TA$)+1):IF TT$<>
"" THEN BP=1:GOTO 3400 ELSE 356
0
3470 IF P THEN LPRINT L1$;GP$;
ELSE PRINT
```



```
3480 PP=0:LC=LC+1:GOSUB 3630:TT
$=MID$(TT$,2):GOTO 3320
3490 TT$=MID$(TT$,2):IF P THEN
LPRINT L1$;GP$;STRING$(TW/4,32)
; ELSE PRINT:LOCATE TW/4
3500 PP=INT(TW/4):LC=LC+1:GOSUB
 3630:GOTO 3320
3510 TT$=MID$(TT$,2):IF LEN(TT$
)>TW THEN PRINT:PRINT"Linha mui
to grande para centralizar:":PR
INTTT$:GOTO 3610
3520 S$=STRING$(INT((TW-LEN(TT$
))/2),32):IF P THEN LPRINT L1$;
GP$;S$;TT$;L1$;GP$; ELSE PRINT:
PRINTS$;TT$
3530 PP=0:LC=LC+1:GOSUB 3630:GO
TO 3560
3540 IF P THEN LPRINT A$;MID$(T
T$,2); ELSE PRINTA$;MID$(TT$,2)
;
3550 PP=0:LC=LC+1:GOSUB 3630
3560 K=K+1:IF P=0 THEN FOR Z=1
TO 500:NEXT
3570 IF K<TL THEN 3290
3580 IF P THEN LPRINT LF$;LF$ E
LSE PRINT:PRINT
3590 IF KF=2 THEN CLOSE#1
3600 IF NOT P THEN 3190 ELSE RE
TURN
3610 PLAY "03L3CR64L3CR64L8CR64
L3C":FOR I=1 TO 2500:NEXT:IF KF
=2 THEN CLOSE#1
3620 RETURN
3630 IF LC>TH THEN IF P THEN LP
RINT LF$;LF$;GP$ ELSE PRINTLF$;
LF$;
3640 LC=1:RETURN
5000 L=CP:LOCATE 0,19:LINEINPUT
 "Procura por: ";TG$
5010 IF TG$="" THEN 5050
5020 PRINT"Procurando..."
5030 IF L=TL THEN CP=TL:CLS:GOS
UB 2080:RETURN
5040 IF INSTR(TX$(L),TG$)=0 THE
N L=L+1:GOTO 5030
5050 CP=L+1:CLS:GOSUB 2080:RETU
RN
5060 IF SS>SE THEN SWAP SS,SE
5070 SE=SE-1
5080 LOCATE 0,20:PRINT"Ordenand
o..."
5090 FOR I=SS TO SE-1
5100 K=I
5110 FOR J=I+1 TO SE
5120 IF TX$(J)<TX$(K) THEN K=J
5130 NEXT:IF I<>K THEN SWAP TX$
(K),TX$(I)
5140 NEXT:CLS:GOSUB 2080:RETURN
```

A rotina de ordenação é acessada a partir do modo editor. Leve o marcador para um dos extremos das linhas a serem ordenadas e tecle **CTRL-O**. A seguir, transfira o cursor para o outro extremo e tecle novamente **CTRL-O**. A ordenação se iniciará imediatamente.

A rotina para a procura de palavras também é ativada dentro do modo editor. Leve o marcador para a posição a partir da qual quer iniciar a busca. Tecle **CTRL-P** e digite a palavra que deve ser procurada. Tecle **[RETURN]** para iniciar a busca. Assim que a expressão for encontrada, o marcador será colocado abaixo da linha que a contém. Para continuar procurando outras ocorrências da mesma palavra, tecle **CRTL-P**; digite **[RETURN]** apenas quando a palavra for solicitada.

Para acessar a rotina de impressão, tecle I a partir do menu principal. Várias perguntas serão feitas. A primeira refere-se à possibilidade do texto a ser impresso vir da memória ou de um arquivo em fita.

A seguir, você deve responder se quer usar "máscara de texto" ou não. Essa rotina possibilita a inserção de partes variáveis no texto principal. Para isso, coloca-se pares de colchetes ("][") nas posições desejadas. Desse modo, quando o programa encontrar esse sinal no texto, irá substituí-lo por um bloco que pode ser carregado do teclado ou de um arquivo. Se você optar por dar entrada pelo teclado, deve digitar o texto necessário toda vez que for solicitado. Caso contrário, a informação será lida de um arquivo em fita (este deverá ter sido previamente preparado para isso).

Antes de iniciar a impressão, é possível também alterar a configuração da impressora. Os valores habituais são: 80 para a largura do formulário, 60 para a largura da linha de texto, 66 para o número total de linhas por formulário e 60 para o número de linhas de texto por página.

Os sinais para formatar o texto são nossos conhecidos: o asterisco ("*") centraliza o texto que o segue; o ampersand ("&") provoca um avanço de linha, de maneira que o texto subseqüente seja iniciado na primeira posição da linha seguinte; já o cifrão ("$"), além de fazer isso, provoca um recuo no texto; finalmente, o sustenido ("#") faz com que a linha seja impressa na margem direita da página.

Quando uma palavra não couber no trecho final de uma linha, deve-se evitar quebrá-la, para que ela não fique dividida por um espaço. O procedimento correto, neste caso, é digitar a palavra inteira na linha seguinte.

```
3000  HOME :BL$ = "IMPRIMIR": G
OSUB 250
3010  IF TL < 2 THEN 3060
3020  PRINT : PRINT : PRINT "IM
PRIME TEXTO DA [M]EMORIA OU [D]
ISCO? ": PRINT "TECLE <CR> PARA
 RETORNAR AO MENU ";
3030  GET R$: IF R$ =  CHR$ (13
) THEN  RETURN
3040  IF R$ <  > "M" AND R$ <
> "D" THEN 3030
3050  PRINT R$: IF R$ = "M" THE
N 3070
3060  GOSUB 4500: HOME :BL$ = "
IMPRIMIR": GOSUB 250
3070  IF TL = 1 THEN  PRINT  CH
R$ (7);: VTAB 12: HTAB 8: PRINT
 "NAO HA TEXTO NA MEMORIA!": FO
R I = 1 TO 1000: NEXT : PRINT
CHR$ (7);: RETURN
3080 KF = 0: HTAB 1: VTAB 9: PR
INT "QUER USAR MASCARA? ";
3090  GET R$: IF R$ <  > "S" AN
D R$ <  > "N" THEN 3090
3100  PRINT R$: IF R$ = "N" THE
N 3220
3110  PRINT : INPUT "QUANTOS BL
OCOS SERAO USADOS? (MAX 20) ";N
B: IF NB < 0 OR NB > 20 THEN 31
10
3120  PRINT : PRINT "OS BLOCOS
SERAO CARREGADOS DO": PRINT "[T
]ECLADO OU [D]ISCO? ";
3130  GET R$: IF R$ <  > "T" AN
D R$ <  > "D" THEN 3130
3140  PRINT R$:BL = 0:KF = 2: I
F R$ = "T" THEN KF = 1: GOTO 32
10
3150  PRINT : INPUT "NOME DO AR
QUIVO MASCARA? ";VB$
3160  IF  ASC (VB$) < 65 OR  AS
C (VB$) > 90 THEN 3150
3170 VB$ = VB$ + ".TXT": PRINT
: PRINT D$;"OPEN";VB$;",S";L1;"
,D";L2
```

```
3180  PRINT D$;"READ";VB$: INPU
T DV,DV
3190  IF DV < NB THEN  PRINT D$
;"CLOSE":ER = 2: GOTO 3580
3200  FOR I = 1 TO NB: INPUT RP
$(I): NEXT : PRINT D$;"CLOSE":
GOTO 3220
3210  PRINT : PRINT : FOR I = 1
 TO NB: PRINT "BLOCO VARIAVEL "
;I;"=>";: INPUT RP$(I): NEXT
3220  HOME : GOSUB 250: VTAB 5:
 PRINT "QUER MUDAR A CONFIGURAC
AO DA IMPRESSORA? ";
3230  GET R$: IF R$ <  > "S" AN
D R$ <  > "N" THEN 3230
3240  PRINT R$: IF R$ = "S" THE
N  GOSUB 5500: HOME :BL$ = "IMP
RIMIR": GOSUB 250
3250 VB = 0:PP = 0:AS = 0:LC =
1: PRINT : PRINT : PRINT "QUER
UMA DEMONSTRACAO NA TELA? ";
3260  GET R$: IF R$ <  > "S" AN
D R$ <  > "N" THEN 3260
3270  PRINT R$:P = 0:G1 = 0: IF
 R$ = "N" THEN P = 1:G1 = GP
3280  FOR K = 1 TO TL - 1: IF
LEFT$ (TX$(K),1) = "#" AND  LEN
 (TX$(K)) - 1 > AS THEN AS =  L
EN (TX$(K))
3290  NEXT : IF AS > TW THEN ER
 = 1: GOTO 3580
3300 K = 1: PRINT : PRINT D$;"P
R#";P: PRINT LF$; SPC( G1);:A$
= "": IF AS > 0 THEN A1 = GP +
TW - AS
3310 TT$ = TX$(K)
3320  IF TT$ = " " THEN  PRINT
 CHR$ (13); SPC( G1);:PP = 0:LC
 = LC + 1: GOSUB 3640: GOTO 354
0
3330  IF KF = 0 THEN 3380
3340  FOR I = 1 TO  LEN (TT$) -
 1: IF  MID$ (TT$,I,2) <  > "]["
 THEN  NEXT :BP = 0: GOTO 3380
3350 BP = I:BL = BL + 1
3360  IF BP = 1 THEN TT$ = RP$(
BL) +  MID$ (TT$,3):. GOTO 3380
3370 TT$ =  LEFT$ (TT$,BP - 1)
+ RP$(BL) +  MID$ (TT$,BP + 2):
 GOTO 3340
3380 TT =  ASC (TT$): IF TT = 3
5 OR TT = 36 OR TT = 38 OR TT =
 42 THEN 3460
3390  IF PP +  LEN (TT$) <  = T
W THEN  PRINT TT$; CHR$ (32);:P
P = PP +  LEN (TT$) + 1: GOTO 3
540
3400 TA$ =  LEFT$ (TT$,TW - PP)
3410  IF  RIGHT$ (TA$,1) = " "
THEN 3440
3420  IF  LEN (TA$) > 1 THEN TA
$ =  LEFT$ (TA$, LEN (TA$) - 1)
: GOTO 3410
3430 TA$ = ""
3440  PRINT TA$; CHR$ (13); SPC
( G1);:PP = 0:LC = LC + 1: GOSU
B 3640:TT$ =  MID$ (TT$, LEN (T
A$) + 1): IF TT$ = "" THEN 3540
3450  GOTO 3390
3460  FOR I = 1 TO 4: IF  MID$
("&$*#",I,1) <  > LEFT$ (TT$,1
) THEN  NEXT
3470  ON I GOTO 3480,3490,3510,
3530
3480  PRINT  CHR$ (13); SPC( G1
);:PP = 0:LC = LC + 1: GOSUB 36
40:TT$ =  MID$ (TT$,2): GOTO 33
30
3490 TT$ =  MID$ (TT$,2): PRINT
  CHR$ (13); SPC( G1); SPC(  IN
T (TW / 4));:PP =  INT (TW / 4)
3500 LC = LC + 1: GOSUB 3640: G
OTO 3330
3510 TT$ =  MID$ (TT$,2): IF  L
EN (TT$) > TW THEN ER = 3: GOTO
 3580
3520  PRINT  CHR$ (13); SPC( G1
); SPC(  INT ((TW -  LEN (TT$))
 / 2));TT$; CHR$ (13); SPC( G1)
;:PP = 0:LC = LC + 1: GOSUB 364
0: GOTO 3540
3530  PRINT  CHR$ (13); SPC( A1
); MID$ (TT$,2);:PP = 0:LC = LC
 + 1: GOSUB 3640: GOTO 3540
3540 K = K + 1: IF P = 0 THEN
FOR Z = 1 TO 1000: NEXT
3550  IF K < TL THEN 3310
3560  IF P = 1 THEN  PRINT LF$;
LF$: PRINT D$;"PR#0": RETURN
3570  IF P = 0 THEN  PRINT : PR
INT : GOTO 3250
3580  PRINT : PRINT D$;"PR#0":
PRINT  CHR$ (7);
3590  HOME : VTAB 12: HTAB 10:
ON ER GOTO 3600,3610,3620
3600  PRINT "ENDERECO MUITO LON
GO!": GOTO 3630
3610  PRINT "NAO HA BLOCOS SUFI
CIENTES": PRINT  TAB( 12)"NO AR
QUIVO ESCOLHIDO!": GOTO 3630
3620  PRINT "IMPOSSIVEL CENTRAL
IZAR!"
3630  FOR I = 1 TO 2500: NEXT :
 PRINT  CHR$ (7);: RETURN
3640  IF LC > TH THEN  PRINT LF
$;LF$; SPC( G1);:LC = 1
3650  RETURN
5200 L = CP:TG$ = "": VTAB 20:
HTAB 1: PRINT "PROCURA POR: ";
5210  GET TC$: IF TC$ <  > CHR
$ (13) THEN TG$ = TG$ + TC$: PR
INT TC$;: GOTO 5210
5220  IF TG$ = "" THEN 5270
5230  HTAB 1: PRINT "PROCURANDO
...";: CALL  - 958
5240  IF L = TL THEN CP = TL: G
OTO 5270
5250  FOR I = 1 TO  LEN (TX$(L)
) - 1: IF  MID$ (TX$(L),I, LEN
(TG$)) <  > TG$ THEN  NEXT :L =
 L + 1: GOTO 5240
5260 CP = L + 1
5270  HOME : GOSUB 2090: RETURN
5300  IF SS > SE THEN TT = SS:S
S = SE:SE = TT
5310 SE = SE - 1
5320  HTAB 1: VTAB 20: PRINT "O
RDENANDO..."
5330  FOR I = SS TO SE - 1
5340 K = I
5350  FOR J = I + 1 TO SE
5360  IF TX$(J) < TX$(K) THEN K
 = J
5370  NEXT : IF I <  > K THEN T
T$ = TX$(K):TX$(K) = TX$(I):TX$
(I) = TT$
5380  NEXT : HOME : GOSUB 2090:
 RETURN
```

Para acessar as rotinas de ordenação e busca de palavras, você deve estar no modo editor.

Se você quiser ordenar um bloco de dados, leve o marcador "**>**" para um dos extremos do bloco e tecle **CTRL-O**. Em seguida, mova-o para o outro extremo e tecle novamente **CTRL-O**. A ordenação começará imediatamente.

Quando for preciso procurar uma determinada palavra dentro do texto, leve o marcador ao ponto escolhido. A seguir tecle **CTRL-P** e digite a palavra desejada, respondendo à solicitação. Tecle **[CR]** e a busca terá início. Quando a palavra for encontrada, o marcador será posicionado logo abaixo da linha que a contém. Para continuar procurando novas ocorrências dessa palavra, repita toda a operação.

Ao mesmo tempo, para imprimir o texto, tecle I a partir do menu principal. Em seguida, responda às perguntas que lhe são apresentadas.

A opção usar "máscara de texto" permite acrescentar ao texto principal blocos variáveis, com os quais podem ser produzidas, por exemplo, cartas individualizadas. Se você fizer essa opção, deixe nos locais apropriados do texto um sinal "][" para cada bloco. Esses sinais serão substituídos pelo texto adequado no momento da impressão.

O texto pode ser digitado a partir do teclado, ou lido de um arquivo em disco. No caso do teclado, o programa solicitará que você digite a frase à medida que os sinais sejam encontrados no texto. Se os blocos forem carregados do disco, o computador lerá um arquivo seqüencial que você deverá ter criado com os dados necessários, usando o próprio programa (se você quiser aproveitar um arquivo já existente, deve acrescentar ".TXT" ao nome dele). Cuide para que os dados estejam na ordem correta e para que sejam suficientes para todas as solicitações.

O texto pode ser formatado com os sinais habituais. (Eles devem sempre ser colocados na primeira posição da linha.) O asterisco ("*") centraliza o conteúdo da linha; o ampersand ("&") provoca um avanço de linha da impressora, colocando o texto subseqüente a partir do início da próxima linha; o cifrão ("$") tem a mesma função, com a diferença de que provoca um recuo no começo do texto; o sustenido ("#") faz as linhas que o contêm serem colocadas à margem direita do texto.

Evite inserir blocos variáveis nas linhas em que aparece o sustenido.

CLASSIFICAÇÃO DOS MICROCOMPUTADORES POR LINHA

LINHA	FABRICANTE	MODELO
Apple II +	Appletronica	Thor 2010
Apple II +	CCE	MC-4000 Exato
Apple II +	CPA	Absolutus
Apple II +	CPA	Polaris
Apple II +	Digitus	DGT-AP
Apple II +	Dismac	D-8100
Apple II +	ENIAC	ENIAC II
Apple II +	Franklin	Franklin
Apple II +	Houston	Houston AP
Apple II +	Magnex	DM II
Apple II +	Maxitronica	MX-2001
Apple II +	Maxitronica	MX-48
Apple II +	Maxitronica	MX-64
Apple II +	Maxitronica	Maxitronic I
Apple II +	Microcraft	Craf II Plus
Apple II +	Milmar	Apple II Plus
Apple II +	Milmar	Apple Master
Apple II +	Milmar	Apple Senior
Apple II +	Omega	MC-400
Apple II +	Polymax	Maxxi
Apple II +	Polymax	Poly Plus
Apple II +	Spectrum	Microengenho I
Apple II +	Spectrum	Spectrum ed
Apple II +	Suporte	Venus II
Apple II +	Sycomig	SIC I
Apple II +	Unitron	AP II
Apple II +	Victor do Brasil	Elppa II Plus
Apple II +	Victor do Brasil	Elppa Jr.
Apple IIe	Microcraft	Craft IIe
Apple IIe	Microdigital	TK-3000 IIe
Apple IIe	Spectrum	Microengenho II
MSX	Gradiente	Expert GPC-1
MSX	Sharp	Hotbit HB-8000
Sinclair Spectrum	Microdigital	TK-90X
Sinclair Spectrum	Timex	Timex 2000
Sinclair ZX-81	Apply	Apply 300
Sinclair ZX-81	Engebras	AS-1000
Sinclair ZX-81	Filcres	NEZ-8000
Sinclair ZX-81	Microdigital	TK-82C
Sinclair ZX-81	Microdigital	TK-83
Sinclair ZX-81	Microdigital	TK-85
Sinclair ZX-81	Prologica	CP-200
Sinclair ZX-81	Ritas	Ringo R-470
Sinclair ZX-81	Timex	Timex 1000
Sinclair ZX-81	Timex	Timex 1500
TRS-80 Mod. I	Dismac	D-8000
TRS-80 Mod. I	Dismac	D-8001/2
TRS-80 Mod. I	LNW	LNW-80
TRS-80 Mod. I	Video Genie	Video Genie I
TRS-80 Mod.III	Digitus	DGT-100
TRS-80 Mod.III	Digitus	DGT-1000
TRS-80 Mod.III	Kemitron	Naja 800
TRS-80 Mod.III	Prologica	CP-300
TRS-80 Mod.III	Prologica	CP-500
TRS-80 Mod.III	Sysdata	Sysdata III
TRS-80 Mod.III	Sysdata	Sysdata Jr.
TRS-80 Mod.IV	Multix	MX-Compacto
TRS-80 Mod.IV	Sysdata	Sysdata IV
TRS-Color	Codimex	CS-6508
TRS-Color	Dynacom	MX-1600
TRS-Color	LZ	Color 64
TRS-Color	Microdigital	TKS-800
TRS-Color	Prologica	CP-400

CLASSIFICAÇÃO DOS MICROCOMPUTADORES POR NOME

FABRICANTE	MODELO	PAÍS	LINHA
Appletronica	Thor 2010	Brasil	Apple II +
Apply	Apply 300	Brasil	Sinclair ZX-81
CCE	MC-4000 Exato	Brasil	Apple II +
CPA	Absolutus	Brasil	Apple II +
CPA	Polaris	Brasil	Apple II +
Codimex	CS-6508	Brasil	TRS-Color
Digitus	DGT-100	Brasil	TRS-80 Mod.III
Digitus	DGT-1000	Brasil	TRS-80 Mod.III
Digitus	DGT-AP	Brasil	Apple II +
Dismac	D-8000	Brasil	TRS-80 Mod. I
Dismac	D-8001/2	Brasil	TRS-80 Mod. I
Dismac	D-8100	Brasil	Apple II +
Dynacom	MX-1600	Brasil	TRS-Color
ENIAC	ENIAC II	Brasil	Apple II +
Engebras	AS-1000	Brasil	Sinclair ZX-81
Filcres	NEZ-8000	Brasil	Sinclair ZX-81
Franklin	Franklin	USA	Apple II +
Gradiente	Expert GPC1	Brasil	MSX
Houston	Houston AP	Brasil	Apple II +
Kemitron	Naja 800	Brasil	TRS-80 Mod.III
LNW	LNW-80	USA	TRS-80 Mod. I
LZ	Color 64	Brasil	TRS-Color
Magnex	DM II	Brasil	Apple II +
Maxitronica	MX-2001	Brasil	Apple II +
Maxitronica	MX-48	Brasil	Apple II +
Maxitronica	MX-64	Brasil	Apple II +
Maxitronica	Maxitronic I	Brasil	Apple II +
Microcraft	Craft II Plus	Brasil	Apple II +
Microcraft	Caft IIe	Brasil	Apple IIe
Microdigital	TK-3000 IIe	Brasil	Apple IIe
Microdigital	TK-82C	Brasil	Sinclair ZX-81
Microdigital	TK-83	Brasil	Sinclair ZX-81
Microdigital	TK-85	Brasil	Sinclair ZX-81
Microdigital	TK-90X	Brasil	Sinclair Spectrum
Microdigital	TKS-800	Brasil	TRS-Color
Milmar	Apple II Plus	Brasil	Apple II +
Milmar	Apple Master	Brasil	Apple II +
Milmar	Apple Senior	Brasil	Apple II +
Multix	MX-Compacto	Brasil	TRS-80 Mod.IV
Omega	MC-400	Brasil	Apple II +
Polymax	Maxxi	Brasil	Apple II +
Polymax	Poly Plus	Brasil	Apple II +
Prologica	CP-200	Brasil	Sinclair ZX-81
Prologica	CP-300	Brasil	TRS-80 Mod.III
Prologica	CP-400	Brasil	TRS-Color
Prologica	CP-500	Brasil	TRS-80 Mod.III
Ritas	Ringo R-470	Brasil	Sinclair ZX-81
Sharp	Hotbit HB-8000	Brasil	MSX
Spectrum	Microengenho I	Brasil	Apple II +
Spectrum	Microengenho II	Brasil	Apple IIe
Spectrum	Spectrum ed	Brasil	Apple II +
Suporte	Venus II	Brasil	Apple II +
Sycomig	SIC I	Brasil	Apple II +
Sysdata	Sysdata III	Brasil	TRS-80 Mod.III
Sysdata	Sysdata IV	Brasil	TRS-80 Mod.IV
Sysdata	Sysdata Jr.	Brasil	TRS-80 Mod.III
Timex	Timex 1000	USA	Sinclair ZX-81
Timex	Timex 1500	USA	Sinclair ZX-81
Timex	Timex 2000	USA	Sinclair Spectrum
Unitron	AP II	Brasil	Apple II +
Victor do Brasil	Elppa II Plus	Brasil	Apple II +
Victor do Brasil	Elppa Jr.	Brasil	Apple II +
Video Genie	Video Genie I	USA	TRS-80 Mod. I

UM LOGOTIPO PARA CADA MODELO DE COMPUTADOR

INPUT foi especialmente projetado para microcomputadores compatíveis com as sete principais linhas existentes no mercado.
Os blocos de textos e listagens de programas aplicados apenas a determinadas linhas de micros podem ser identificados por meio dos seguintes símbolos:

Sinclair ZX-81

TRS-80

TK-2000

MSX

Spectrum

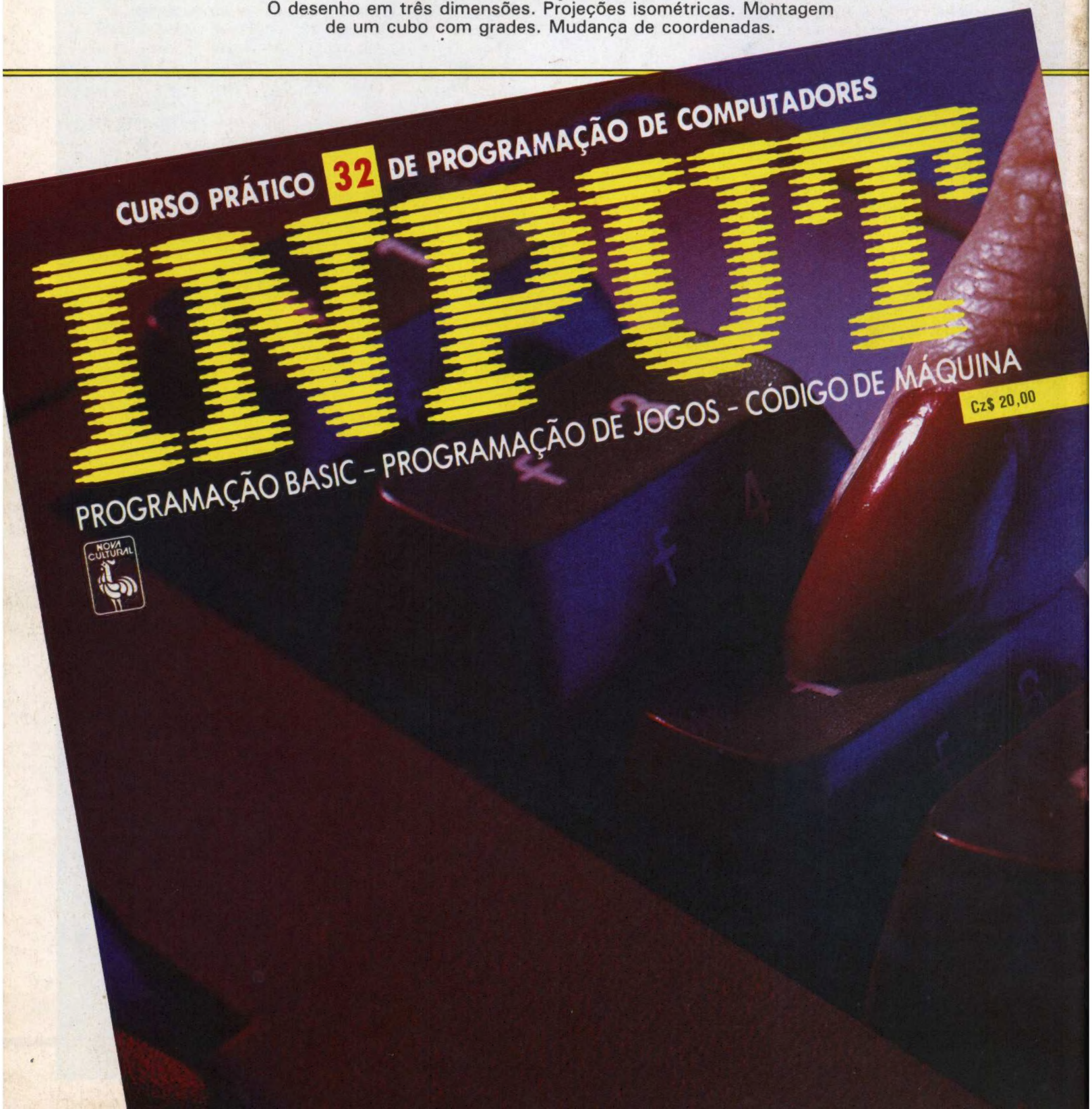
TRS-Color

Apple II

Quando o emblema for seguido de uma faixa, então tanto o texto como os programas que se seguem passam a ser específicos para a linha indicada.

NO PRÓXIMO NÚMERO

PROGRAMAÇÃO BASIC
Sejam quais forem os seus dados, será mais fácil entendê-los se você organizá-los em gráficos de barras e segmentos.

PROGRAMAÇÃO BASIC
Teclas programáveis: um recurso fantástico oferecido pelo MSX para dar aos programas um acabamento impecável.

PROGRAMAÇÃO BASIC
O desenho em três dimensões. Projeções isométricas. Montagem de um cubo com grades. Mudança de coordenadas.